SAPIENZA
UNIVERSITÀ DI ROMA

Generative Models for Inference: An Application to Authorship Attribution.

Scuola di dottorato Vito Volterra

Dottorato di Ricerca in Fisica – XXXIV Ciclo

Candidate

Giulio Tani Raffaelli

ID number 1407269

Thesis Advisors

Prof. Vittorio Loreto

Dr. Francesca Tria

April 2022

Thesis defended on 26th May 2022
in front of a Board of Examiners composed by:

Prof. Roberto Di Leonardo (chairman)

Prof. Roberto Benzi

Dr. Josè Lorenzana

External referees:

Dr. Vito D.P. Servedio

Dr. Bernat Corominas Murtra

Generative Models for Inference:An Application to Authorship Attribution.
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: 20th April 2022

Author's email: giulio.tani@uniroma1.it

A Iza

Abstract

Computer-aided stylometry is a powerful tool in authorship attribution. Recent models can point the author of an anonymous text among thousands or distinguish different contributors to one text. However, most methods are quite complex and depend on the language. We propose a new Authorship Attribution method based on inference using a stochastic process. Every author is associated with the process that is most likely to reproduce their known corpus. We assign a text to the author whose process gives the highest probability of producing the text. We find high attribution rates independent of the language of the text or the tokenisation. Inference using stochastic processes offers exciting opportunities for stylometry and information retrieval.

Thanks to my thesis advisors for their guidance in these three years. Thanks to Alexandra Elbakyan, whose help was fundamental in the study and documentation for this thesis. Thanks to MVP. She does not know, but she saved this thesis in the last weeks.

Contents

Introduction

Finding the author of anonymous texts has been a challenge for centuries. Since the beginning of the modern era [151], scholars have studied texts of debated attribution. However, for many centuries this was based on qualitative analysis. Scholars had to evaluate whether an author would use some word or a piece of text is in their style. This approach led to exciting and revolutionary discoveries but could not quantify its findings.

Stylometry brought a revolution into the field. Around the beginning of the twentieth-century [93, 105], scholars started to analyse the frequency of the words in texts. These frequencies could indicate whether an author would use some specific words. It became possible to describe and measure an author's style through a consistent analysis of their known corpus. This allowed for the first time to measure the internal differences in an author's production or select the most likely author of a text out of a set of candidates.

The introduction of computers brought a new significant improvement. In the sixties [106] analysing big corpora became a matter of minutes. Scholars could spend more time designing refined techniques instead of counting words. The methods became more effective and grew in number and complexity. Some techniques are general and applicable to any text. Some are specific for a genre or a language. The analysis may consider groups of characters, words, syntactic relations, and the very meaning of words.

Many techniques look for an author's fingerprint in commonly repeated words [129]. They often rely on lists of most used words, ranging from the tens [129] to the thousands [142]. Words in these lists are the only to be considered [129, 142, 139] or removed [7] from the texts. Other techniques rely on large databases to bring words to their base form (e.g. infinitive form of verbs and singular of nouns) or to tag their grammatical function [7, 86].

All these techniques have a degree of subjectivity in creating lists and databases and are useful only for a specific language. A different class of methods is language independent and uses character N-grams [80], word frequencies or N-grams [142],

topic models [129, 135] or even – following an information theory approach – looks at the authors as information sources [89, 15, 69, 67].

In contrast, we follow a different approach. We model the sequence of words or characters in a text as a stochastic process. Our approach applies inference using Poisson-Dirichlet processes to determine the author of an anonymous text. Given a Poisson-Dirichlet process, we map the sequence of words or strings in a text to samples from the process.

A Poisson-Dirichlet process is a stochastic process whose realisations are discrete probability distributions. Every sample from the process may repropose already seen elements or introduce a new element drawn from a base probability distribution. Every innovation fosters the introduction of more innovations.

We model the text itself as a sequence whose tokens are samples from a PD process. We determine the probability of having the sequence of an unknown text being the following output of the same process that generated all the texts of a known author.

Our approach requires a minimal training phase and depends only on a small set of hyperparameters. These parameters influence the variables used, the base probability distribution of the Poisson-Dirichlet processes, and how we compare each text to each author.

Besides being simple and straightforward, this technique offers state of the art results. We were able to reliably find the author of a book among tens to tens of thousands of alternatives. Moreover, this result is substantially independent of the language of the text. The PD process proves to be a good model for inference on texts.

Chapter 1

Generative Models, Stochastic Processes and Inference

This thesis will apply inference using generative models to tackle a common stylometry task. Before entering the details of stylometry (see chapter 2) and our specific approach (see chapter 3), we will describe generative models. In particular, we will focus on processes featuring innovation and show how it is possible to apply them to inference.

1.1 Generative Models

In a few words, a generative model is a statistical model for some joint probability distribution. It is possible to generate, i.e. draw, samples from this distribution, hence the name. If the distribution adapts well to mimic some population, the generated samples seem to come from the population itself.

Famous examples of generative models are the recent breakthrough in Generative Adversarial Networks (GAN). These networks can produce fake pictures of nonexistent people who seem perfectly real at first, and often also second, look[1]. These models do not output a probability explicitly. However, they encode information like that a person usually has only one mouth but two eyes or that, if they are wearing an earring, they are probably wearing two.

We will set aside GAN, and Artificial Neural Networks (ANN) in general. These models are widely used and find applications also in stylometry (see section 2.1.5). However, we prefer proper statistical generative models that output well-defined probabilities through transparent procedures. This preference is not a whim: in

[1] Try, for example: https://this-person-does-not-exist.com/en, last checked January 12, 2022.

many applications, from humanities to court trials, explaining the reason for the output is fundamental.

We will focus our study on statistical models. In particular, we will focus on generative models where the probability of the next element does not depend on future elements. However, we are not assuming that every text is a stream of consciousness. Humans conceive texts having in mind their future elements. For example, the content of these paragraphs depends on the following chapters. Choosing these models requires that the order of the elements in a text allows understanding without future information.

When working with natural language processing (NLP), the most common class of generative models are token N-gram models. The principle of these models is straightforward: the probability of a token depends on its context, but we assume that the last $N-1$ tokens are enough to determine the probability of the next one. In formulas:

$$\begin{aligned} P(w_n, w_{n-1}, \dots, w_1) = P(w_n \mid w_{n-1}, \dots, w_1) P(w_{n-1}, \dots, w_1) \sim \\ \sim P(w_n \mid w_{n-1}, \dots, w_{n-(N-1)}) P(w_{n-1}, \dots, w_1) \end{aligned} \quad (1.1)$$

By the chain rule, we can then express every token's conditional probability as a function of the former $N-1$.

This approach looks fine, but what is the origin of the conditional probabilities? We estimate the probabilities with the N-gram frequency in the training corpus. This estimate creates a problem when a new N-gram appears. A new N-gram has past frequency precisely zero, and this would imply zero probability to the generated sequence.

Usual approaches use interpolation or a back-off estimator to avoid zero probability N-grams. They resort to shorter N-grams. Indeed, it is more likely to observe the shorter sequence $w_{n-(N-2)}, \dots, w_n$ than $w_{n-(N-1)}, \dots, w_n$, and its frequency can be the base for the desired probability. Then, probabilities must be normalised, and many smoothing approaches exist. If the shorter N-gram is also missing, we need to repeat this procedure until we get an estimate of the probability.

Except for a few examples[2], these approaches do not include a probability for novel elements from first principles. This lack is a limitation because the smoothing or interpolation are approximate procedures, and the model often has to rely on them. Indeed, because of the fat-tailed distribution of words, the fraction of tokens appearing only once (*hapax legomena*) in a text can be pretty high, up to 40% [129]

[2]It is possible [148] to interpret Interpolated Kneser-Ney [79] as an approximate inference method in a Bayesian model using a Poisson-Dirichlet process (see section 1.2). This interpretation, however, came eleven years after the proposal of the method.

and above. There are good chances that a new document will include many unseen tokens, let alone unseen N-grams.

A different approach would be to use text models explicitly designed to model the occurrence of new elements.

Some Useful Scaling Laws

Before entering into the detail of the different models, we need some tools to evaluate them. We will compare their ability to reproduce observable features from systems like, in this case, texts. We will focus on three different scaling laws observed in texts that also enjoy a certain degree of universality. These laws are known as the Zipf's, Heaps' and Taylor's laws.

Zipf's Law This law describes a power-law relation between the frequency f and the rank R of words. The words are ordered in a list with the most frequent first. J.-B. Estoup [48] observed it for the first time in texts, and later, Zipf [163] rediscovered it. Frequencies following a Zipfian law obey the rule:

$$f \propto R^{-\alpha} \tag{1.2}$$

In the original work, Zipf found $\alpha \simeq 1$. This exponent is usually observed (in texts) for relatively low ranks, roughly up to the thousands or tens of thousands. For higher ranks – less frequent tokens – the value of α is greater than 2. Over the years, researchers found many similar power-law behaviours with various exponent values. All these laws are collectively called Zipf's laws.

The use of the Zipf's law is widespread as it is possible to appreciate it on a simple log-log plot of the frequency versus the rank. Zipf's law is a consequence of the power-law distribution of the frequencies. Zipf's law holds if the number of elements with a given frequency is a random variable with power-law distribution:

$$P(f) \propto f^{-1-\frac{1}{\alpha}} \tag{1.3}$$

Any model should be able to reproduce the Zipf's law relation and to adjust the exponent α.

Heaps' Law This law relates the number k of different elements (or types) in a sequence with the number n of elements. Observed by Herdan [60] and later by Heaps [59] respectively in linguistics and information retrieval, it takes the form:

$$k \propto n^{\beta} \tag{1.4}$$

with exponent $\beta \in (0, 1]$. Usually, for short sequences $\beta \simeq 1$ while, for higher values of n, it decreases to values $\beta \approx 0.5$.

Heaps' law is an essential requirement for any model too. It encodes the continued appearance of new elements (no saturation) but at a decreasing pace ($\beta < 1$ in many systems).

A renowned relation between these two laws states that the exponents are related as:

$$\begin{cases} \beta = \frac{1}{\alpha} & \alpha > 1 \\ \beta \simeq 1 & \alpha = 1 \\ \beta = 1 & \alpha < 1 \end{cases} \tag{1.5}$$

To obtain this relation, consider the moment a new element enters the system. It will have frequency $\frac{1}{n}$, and its rank will be equal to the maximum rank, i.e. the number of different elements k. In the case of $\alpha > 1$, the correct normalisation of Zipf's law for large k is independent of the maximum rank k. Thus we have:

$$\begin{cases} \frac{1}{n} \propto k^{-\alpha} & \text{from Zipf's law} \\ k \propto n^{\beta} & \text{from Heaps' law} \end{cases} \tag{1.6}$$

From which the first relation in Eq. (1.5) follows. In the case of $\alpha \leq 1$, the normalisation of Zipf's law depends on the maximum rank k, and the result follows from a similar reasoning[3].

This simple relationship holds only in the limit for $n \to \infty$. Because of finite size effects [92], the measured Heaps' exponent tends to be smaller than expected, and the relationship in Eq. (1.5) appears only on the tails of the data.

Taylor's Law Taylor observed this third law reviewing studies about animal populations [147] as a relation between different system realisations. He noticed that by sampling the population of very different species of animals (from plankton to insects, from worms to fishes), the variance in the number of counts had a strong dependence on the mean. He observed a relation of the form:

$$\sigma \propto \mu^{\gamma} \tag{1.7}$$

and found values ranging from 0.35 to 1.54[4]. In this relation, values of $\gamma > \frac{1}{2}$ mean a tendency of the population to aggregate, $\gamma = \frac{1}{2}$ imply a random distribution and $\gamma < \frac{1}{2}$ is characteristic of species with repulsion interactions and a near-regular spatial distribution.

Researchers found several systems that follow this law in fields as diverse as epidemiology, urbanism, and linguistics, see figure 1.1. On the other hand, as already

[3] In the case $\alpha = 1$, Zipf's law gives $n = k \log k$ thus is not a pure power-law behaviour. However, in practice, the curve is almost linear.

[4] In the original paper, Taylor introduced the relation for σ^2, and all the numbers are doubled.

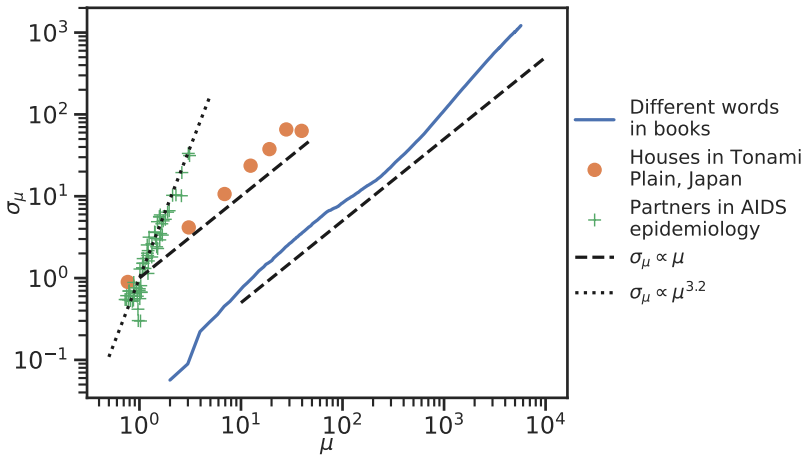


Figure 1.1. Taylor's law in real systems. Green crosses are the variance in reported numbers of different sexual partners versus the mean, from AIDS epidemiology [8]. Orange dots are the variance versus the mean in the number of houses in the Tonami Plain in Japan using squares $100 \times 100\,\mathrm{m}^2$ to $700 \times 700\,\mathrm{m}^2$ from [73]. The blue line is referred to the number of different words in texts as in [149]. The black dashed lines are intended as a guide for the eye.

observed by Taylor, random samples from a distribution give an exponent $\gamma = \frac{1}{2}$. Gerlach and Altmann [56] show that if an independent Poisson process governs the appearance of each element w with a given power-law distributed probability $P(w)$, it produces Heaps' and Zipf's laws with the expected exponents. However, it leads to Taylor's law with exponent $\frac{1}{2}$, i.e. $\sigma \propto \sqrt{\mu}$ as in a central-limit-theorem-like convergence.

Taylor's law is a stronger requirement for a model for texts. Obeying (on average and asymptotically) to Zipf's and Heaps' laws is not enough. The relative fluctuations of the vocabulary size around its mean do not decrease in longer samples. It is a non-self-averaging quantity: deviation from the mean has to grow fast also in models.

The following sections present a review of the models proposed to represent the emergence of the new. Not all the models described can reproduce observed behaviours. However, they all contributed to the field with new concepts or the originality of the approach.

1.1.1 Yule-Simon-like Models

Yule and Simon introduced their model for innovation in 1955 [138]. The observation that the frequency distribution of words in texts written in a given language follows a

fat-tailed distribution has been puzzling the scientific community since the beginning of the 20th century [48, 163] and continues to be debated [51].

Yule-Simon's model cannot generate a sequence of tokens obeying a sub-linear Heaps' law. The token frequency distribution is represented by a power-law with an exponent smaller than 2, see Eq. (1.3). Consequently, the Zipf's exponent $\alpha < 1$ and the associated Heaps' law is linear. The model proposed by Zanette and Montemurro overcomes this limitation, even if the sub-linear Heaps' exponent has to be recovered by data and inserted by hand without a first principle explanation.

Plain Yule-Simon Model The Yule-Simon model generates a stream of tokens according to the following two prescriptions. At the beginning, i.e. at time $t = 1$, only one token is present in the stream. At a generic time t, a new token is added to the stream with probability p, while with complementary probability $(1 - p)$ we choose a token randomly extracting it from the stream. In this way, the tokens that appear more frequently in the stream are more likely to be extracted.

Because of its sequential nature, the Yule-Simon model is particularly suitable for describing linguistics phenomena. However, when used for the generation of texts, some key aspects still cannot be reproduced. Above all, it is evident that the rate of addition of new tokens is constant in time (p), thus resulting in linear growth of the number of different tokens $k = pt$. In contrast, in actual texts, such growth asymptotically follows a sub-linear Heaps' law.

The mechanism of favouring those elements that occur more frequently in the stream is called "*rich-gets-richer*". It is now a paradigm for generating tokens with a power-law frequency distribution. Notably, the preferential attachment rule introduced in the Barabási-Albert model [14] is a form of *rich-gets-richer*.

However, the power-law in the frequency rank is not satisfying. Solving the model for the distribution of elements appearing n times in the sequence q_n, we find a power-law behaviour with $q_n \propto n^{-1-\frac{1}{1-p}}$. A power-law distribution of frequencies with exponent $1 + \frac{1}{1-p}$ corresponds to a frequency-rank exponent $\alpha = (1 - p)$ always smaller than one.

Yule-Simon's Model with Time Dependent Sub-Linear Invention Probability The Yule-Simon model is a good starting point but bears two main issues. First, with p in the range between 0 and 1, it is impossible to recover frequency-rank exponents α larger than 1 although lots of idioms display them. Second, the dictionary, i.e. the number of different tokens, grows linearly in time and not sub-linearly, thus faster than reality.

To correct both issues, Zanette and Montemurro introduced in [159] a time-dependent and decreasing probability $p_t = p_1 t^{\beta-1}$ with $0 < \beta < 1$. This decay assures that the dictionary $k(t)$ will grow as t^β since, by definition, $k(t) = \int_1^t p_1 s^{\beta-1}\, ds$.

This model fixes in a somewhat artificial way the inability of the plain Yule-Simon model to reproduce exponents $\beta < 1$. Due to the link between Heaps' and Zipf's laws, it also shows frequency-rank distribution decreasing as a power-law of exponent $\alpha = \frac{1}{\beta} > 1$ as observed in actual processes.

1.1.2 The Sample-Space Reducing Model

An interesting attempt to explain fat-tailed distributions is the Sampling Space Reducing (SSR) model proposed in [33]. Notably, it tries to reproduce power-law frequency distributions without explicitly resorting to a rich-gets-richer mechanism.

The model catches the idea that the space of possibilities often locally reduces when the process goes on. The first word is almost free when composing a sentence, while the subsequent ones are more and more constrained. The space of the available options reduces while we approach the teapot[5].

This simple process works as follows: (i) the process starts with an N-faced dice; (ii) at time t, we trow a j-faced dice resulting from the evolution of the initial N-faced dice, and let i be the face value obtained; (iii) at time $t+1$ an $i-1$-faced dice is then thrown, and the process goes on until we roll a 1. Independently of N, the visiting probability for the site i, defined as the probability that a particular process visits the site i before ending at 1, is

$$P_N(i) = \frac{1}{i} \tag{1.8}$$

If we consider a cyclic process, the process starts again from an N -faced dice when we roll a one. Then, the visiting probability is also proportional to the occupation probability and thus to the frequency rank, reproducing an exact Zipf law $f(R) \propto R^{-1}$.

The authors also study the case in which a probability λ exists to come back at the N-faced dice at each step. By relaxing the constraint of the sample space constant reduction, the model, named "noisy" in [33], behaves as a superposition of the pure sample-space reducing process (with probability $1-\lambda$) and of a random process where a number is drawn uniformly in the interval $[1, N]$ (with probability λ). The authors show that one obtains in this case a generalised Zipf law with frequency-rank distribution $f(R) \propto R^{-\lambda}$.

[5] The surprise in finding 'teapot' instead of 'end' is a possible sign that, in this context, the space shrunk to include one word but not the other.

In [34], the authors proposed further variant that modifies step (iii). Instead of throwing each time a single dice, now μ dice are thrown. For non-integer μ, it is enough to throw μ dice at each step on average. For $\mu = 1$ the original definition is recovered, with $\mu < 1$ there is, at each step, a probability of stopping the process, i.e. no more dice to throw, and starting again with an N-faced dice. This case is equivalent to the noisy model with $\lambda = \mu$. With $\mu > 1$, the visiting probability of sites with small i increases. While the number on the faces decreases, the number of dice thrown increases. In this case, the probability that a particular process visits the site i before ending at one is

$$P_N(i) = \frac{1}{i^{\,\mu}} \tag{1.9}$$

This model can thus reproduce the full spectrum of exponents in the frequency-rank distribution.

However, the SSR model model cannot reproduce a proper Heaps' law. The number of sites is fixed, and the number of different elements observed saturates to N instead of following a sub-linear power-law behaviour. When the system is far from saturation, i.e. when $k \ll N$, as demonstrated in [97], the growth of k follows a power law with exponent $\gamma = \frac{1}{\mu}$ for $\mu > 1$ and $\gamma = 1$ otherwise. The same paper also shows how this model can reproduce the statistics of shared components [98] observed in natural systems. Having natural labelling for the states independent from the order of appearance is a property regarded as necessary for this to happen [97].

1.1.3 Hoppe Urn Model

Fred M. Hoppe introduced this model in 1984 [62] in genetics. It is an urn model like the renowned Pólya urn but, for the first time, introduces the concept of novelties. It describes the allelic partition of a random sampling of n genes from an infinite population at equilibrium. Under some specific hypotheses[6], the probability of sampling the $(j+1)$-th gene with an allele already sampled is $j/(j+\theta)$, where $\theta = 4N\mu$ (Ewens' sampling formula).

Pólya Urn In the classical version of the Pólya urn model, an urn contains balls of various colours. A ball is drawn at random, inspected, and placed back in the urn along with a certain number of new balls of the same colour, thereby increasing that colour's likelihood of being drawn again in later rounds. The resulting rich-gets-richer dynamics leads to skewed distributions and has been used to model the emergence

[6] The population evolves according to a discrete-time neutral Wright-Fisher process with a constant mutation rate μ per gene. We take the infinite population limit $N \to \infty$ and $\mu \to 0$, with $N\mu$ constant, and sample j genes from the population at equilibrium.

of power laws and related heavy-tailed phenomena in fields ranging from genetics and epidemiology to linguistics and computer science.

Hoppe Urn Hoppe considered a Pólya urn with balls of two different qualities: black balls of mass θ and coloured balls with mass one. The dynamical process works as follows: it starts with only a black ball in the urn, then balls are randomly chosen from the urn proportionally to their mass. At each time step t, (i) if the extracted ball is black, a ball with a brand new colour is added to the urn together with the black ball, (ii) if the extracted ball is coloured, it returns to the urn along with an additional copy of it. It is easy to see that the probability of extracting an already existing colour from the urn at each step $t+1$ is exactly $P_{existing}(t+1) = t/(t+\theta)$, reproducing the result from Ewens.

The expected number of different colours in the urn at step n can be computed explicitly and reads

$$k(t) = \frac{\theta}{\theta} + \frac{\theta}{\theta+1} + \frac{\theta}{\theta+2} + \cdots + \frac{\theta}{\theta+t-1} \tag{1.10}$$

This expression can be approximated as $k(t) = \theta \log(\theta + t)$, from which we can obtain an estimate:

$$f(R) \simeq \frac{t}{\theta} \exp\left(-\frac{R-1}{\theta}\right) \tag{1.11}$$

The results for k and f show that also Hoppe's model is not satisfying. The number of different elements has a sublinear growth but is too slow for any power law. The decay in the frequency-rank distribution is too fast.

1.1.4 Comparison of the Models

None of the models introduced in the previous sections proves to be fully satisfying. Except for the Zanette-Montemurro, where the variation seems too *ad hoc*, Yule-Simon like models fail to reproduce a sublinear growth of the number of different elements and suffer from limitations in the choice of α. The Sample Space Reducing model, even if extended to reproduce every value of α, is limited by the fixed number of possible different elements, and the Heaps law holds only for a limited time. Hoppe's urn model has the sublinear growth missing from Yule-Simon-like models but is too slow for natural systems.

In Table 1.1, we present a scheme summarising the features of the models. Then, numerical simulations of the models of the last sections are presented in Fig. 1.2, showing the power law behaviour (if present) of the different models. Finally, as none is satisfying, we introduce in the next section the Pólya Urn with Triggering. This model offers a way to independently derive Heaps' and Zipf's laws from the idea of the *adjacent possible*.

Table 1.1. Comparison of the behaviours of the models proposed. YS stands for Yule-Simon's; ZM for Zanette-Montemurro; SSR for Sample Space Reducing model; PUT for Polya Urn with Triggering.

Model	**Notes**	**Zipf's exponent α**	**Heaps' exponent γ**
YS	Constant invention rate p	$1-p$	1
ZM	Variable invention rate $p = ct^{\gamma-1}$ with $0 < \gamma < 1$	$\frac{1}{\gamma}$	γ
SSR	$\mu > 1$ dices thrown at each step	μ	$\frac{1}{\mu}$ †
	$\mu < 1$ dices thrown at each step	μ	1 †
Hoppe	No power-law behaviour	$f(R) \simeq \frac{t}{\theta}\exp\left(-\frac{R-1}{\theta}\right)$	$k(t) = \theta\log(\theta+t)$
PUT	$\rho > \nu$	$\frac{\rho}{\nu}$	$\frac{\nu}{\rho}$
	$\rho < \nu$	$\frac{\rho}{\nu}$	1

† Valid only for $k \ll N$.

1.1.5 Urn Model with Triggering

This section presents the Pólya Urn with Triggering (PUT), a generalisation of the urn models seen in the preceding section, presented in [150]. This model incorporates the notion of the *adjacent possible* so that one novelty can trigger further novelties.

Adjacent Possible Stuart Kauffman introduced the concept of *adjacent possible* for biological systems [72]. The *adjacent possible* includes all those elements (bacteria, texts, cellphones) that are just one step (mutation, edit, feature) away from known elements. A central characteristic of the *adjacent possible* is that – when one of its elements becomes actual – it enlarges instead of simply losing one element and shrinking. Indeed, any new element observed introduces a whole new set of elements in the *adjacent possible* that are one step away from it.

The PUT model thus builds on that of Hoppe and other researchers, who introduced novelties within the framework of Pólya's urn. However their models did not posit that novelties could trigger subsequent ones. Hoppe's urn scheme is non-cooperative in the sense that there is no account for the conditional appearance of new colours; in

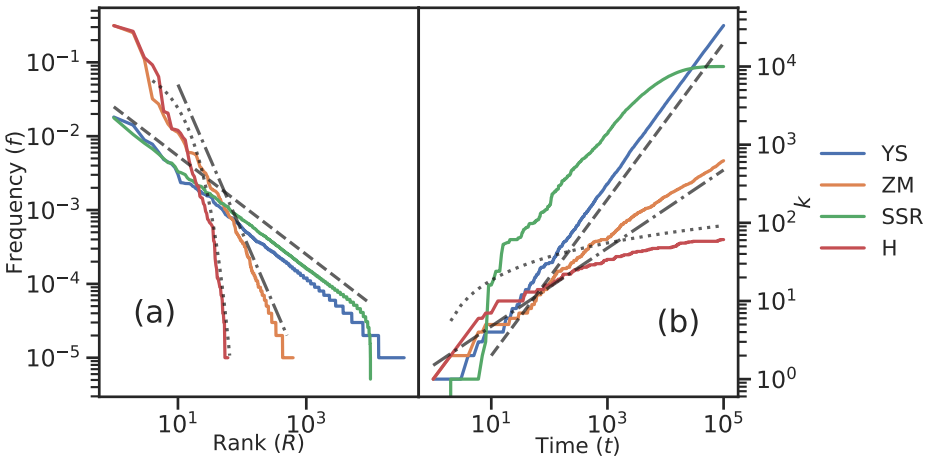


Figure 1.2. Heaps' law (left) and Zipf's law (right) for the models presented. YS stands for Yule-Simon with $p = 1/3$; ZM stands for Zanette-Montemurro with $c = 1$ and $\gamma = 1/2$; SSR stands for Sample Space Reducing model with $N = 10^4$ and $\mu = 2/3$; H stands for the Hoppe model with $\theta = 7$. The black dashed line is a power law with exponent $\gamma = 1$ in the Heaps plot and with $\alpha = 2/3$ in the Zipf plot. The black dot-dashed line is a power law with exponent $\gamma = 1/2$ in the Heaps plot and with $\alpha = 2$ in the Zipf plot. The black dotted line are functions of the kind of those reported in Table 1.1 for the Hoppe model.

particular, one novelty does nothing to facilitate another. In contrast, the cooperative triggering of novelties is essential to the PUT model.

Model Definition

Consider an urn $\mathcal{U}$ initially containing N_0 distinct elements, represented by balls of different colours. We can think of these elements as songs we have listened to, inventions, ideas, words in a text. A series of words (songs, inventions) is idealised in this framework as a sequence $\mathcal{S}$ of elements generated through successive extractions from the urn.

As the *adjacent possible* expands when something novel occurs, the urn's contents enlarge whenever we draw a novel element. When a horse appears for the first time in a novel, we suddenly expect that at some point also riders, saddles, horseshoes, rides and falls will come in. These concepts are related and now accessible since we are on the topic[7].

[7] This kind of relatedness is strong. Finding a spaceship in the middle of the eighteen century would be surprising. Authors such as Douglas Adams value these kinds of odd juxtapositions for their humorous effect:

> [Vogon spaceships] hung in the air in much the same way that bricks don't.

from *The Hitchhiker's Guide to the Galaxy* [2].

Mathematically we consider an ordered sequence $\mathcal{S}$, constructed by picking elements (or balls) from a reservoir (or urn) $\mathcal{U}$, initially containing N_0 distinct elements. According to the following procedure, both the reservoir and the sequence increase their size. At each time step:

1. an element is randomly extracted from $\mathcal{U}$ with uniform probability, its identity recorded in $\mathcal{S}$;
2. the extracted element is put back into $\mathcal{U}$ together with ρ copies of it;
3. if the extracted element has never appeared before in $\mathcal{S}$ (it is a new element in this respect), then $\nu + 1$ different brand new distinct elements are added to $\mathcal{U}$.

Note that the number of elements n of S, i.e. the length $|\mathcal{S}|$ of the sequence, equals the number of times t we repeated the above procedure. If we let k denote the number of distinct elements that appear in S, then the total number of elements in the reservoir after t steps is $|\mathcal{U}|_t = N_0 + (\nu + 1)k + \rho t$.

In this model, a novelty sets the stage for other novelties by triggering new elements in the urn, hence its name Pólya Urn with Triggering.

The generalised Zipf's law, computed for large values of t and R, i.e. low-frequency elements, reads:

$$f(R) \propto R^{-\frac{\rho}{\nu}} \tag{1.12}$$

it is then possible to recover the full spectrum of values for α.

The choice of the ratio ρ/ν influences the Heaps' exponent. When $\nu > \rho$ the number of new, never extracted elements in the urn grows faster than known elements. The probability of finding a new element is not decreasing over time, as in the Yule-Simon model, and the Heaps' exponent is $\beta = 1$. When $\nu < \rho$, the number of known elements grows faster, and the probability of getting a new one decreases over time. This slows down the growth of the number of different elements in the sequence to

$$k \sim (\rho - \nu)^{\frac{\nu}{\rho}} t^{\frac{\nu}{\rho}} \tag{1.13}$$

In this expression, we find that the relation $\beta = 1/\alpha$ is not a trivial consequence, as in the case of sequences obtained sampling a power-law distribution. Here the relationship holds only in the limit of large t and R and is a consequence of the expanding space of possibilities (see Fig. 1.3).

The authors analyse four data sets in [150], each consisting of a sequence of elements ordered in time: (1) Texts: the elements are words. A novelty is defined as the first occurrence of a word in the text; (2) Online music catalogues: the elements are songs. A novelty occurs whenever a user listens to a track that they have not listened to before; (3) Wikipedia: the elements are individual wiki pages. A novelty corresponds to a contributor's first edit action of a wiki page; (4) Social annotation

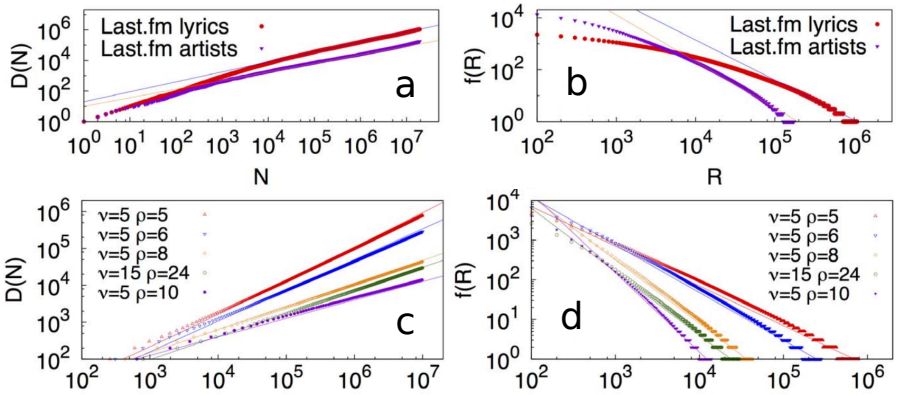


Figure 1.3. Heaps' law (a, c) and Zipf's law (b, d) in the real dataset Last.fm (a) and (b) and in the urn model with triggering (c, d). Straight lines in the Heaps' law plots show functions of the form $f(x) = ax^{\beta}$, with the exponent $\beta = 0.68$ (Last.fm lyrics) and $\beta = 0.56$ (Last.fm artist), and to the ratio ν/ρ in the urn model with triggering, showing that the exponents for the Heaps' law of the model predicted by the analytic results are confirmed in the simulations. Straight lines in Zipf's law plots show functions of the form $f(x) = ax^{-\alpha}$, where the exponent α is equal to β^{-1} for the different β's considered above. Note that the frequency-rank plots in real data deviate from a pure power-law behaviour and the correspondence between the β and α exponents is valid only asymptotically. Figure from [150].

systems: in the so-called tagging sites, the elements are tags (descriptive words assigned to photographs, files, or other pieces of information). A novelty corresponds either to the introduction of a brand new tag (a true innovation), or to its adoption by a given user.

The growth of the number $k(n)$ of distinct elements (words, songs, wiki pages, tags) in a temporally ordered sequence of data of length n quantifies the rate at which novelties occur. Here we report the results only for the Last.fm dataset referring to the original papers for the others. Figure 1.3 (a, c) shows a sub-linear power-law growth of $k(n)$ with exponent $\beta < 1$. This sub-linear growth is the signature of Heaps' law. It implies that the rate at which novelties occur decreases over time as $t^{\beta-1}$.

We look for Zipf's law in the frequency-rank distribution of the elements inside each sequence of data. In all cases (Fig. 1.3 b, d), the tail of the frequency-rank plot also follows an approximate power law (Zipf's law). Moreover, its exponent α is compatible with the measured exponent β of Heaps' law for the same data set, via the relation $\beta = 1/\alpha$.

It is essential to observe that the frequency-rank plots are far from featuring a pure power-law behaviour. In particular, the relation $\beta = 1/\alpha$ between the exponent

β of Heaps' law and the exponent α of Zipf's law is expected to hold only in the tail of the Zipf plot.

Another interesting feature of the PUT model presented in [149] is its ability to reproduce the Taylor law. From numerical simulations, we obtain an exponent $\frac{1}{2}$ with the Yule-Simon and its derived models (like Zanette-Montemurro) and probably with the Hoppe model (the slow logarithmic growth of k over time makes it impossible to verify the exponent for large k). The SSR has an exponent $\approx \frac{1}{2}$ when $k \ll n$ and then, due to the saturation effect, has σ decreasing for large k. PUT model instead reproduces the correct exponent when $\nu < \rho$.

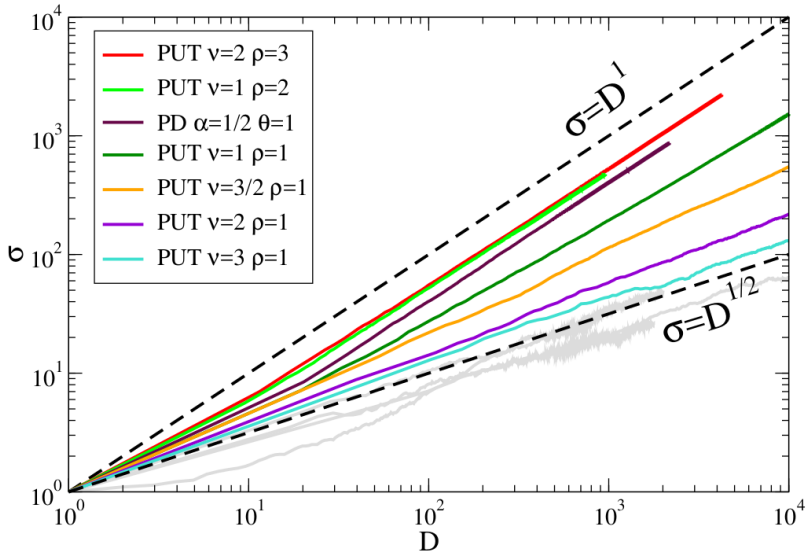


Figure 1.4. Taylor's law for in models.. The grey curves with exponent $\frac{1}{2}$ are for Zanette-Montemurro, Yule-Simon and random sampling from a power-law distribution. PUT curves are for the PUT model and the PD curve is for the Poisson-Dirichlet process, see section 1.2. Figure from [149], the authors call D the number of different elements k.

Extensions to the PUT Model

This simple PUT model can account simultaneously for the emergence of Heaps', Zipf's and Taylor's laws. This is an interesting result *per se* because it offers a possible solution to the longstanding problem of explaining the origin of Heaps' and Zipf's laws through a single microscopic mechanism, without the need of hypothesising one of them to deduce the other.

Despite the interest of this result, this is not yet enough to account for the *adjacent possible* mechanism revealed in actual data. In its present form, the model accounts for the opening of new perspectives triggered by a novelty but does not contain any bias towards realising these new possibilities. To account for this,

in [150] the authors infuse the notion of semantics into the model. They endow each element with a label, representing its semantic group, and allow for the emergence of dynamical correlations between semantically related elements.

This extended model, called semantic PUT (sPUT), captures some of the main qualitative features of clustering seen in actual data (see section Methods of [150]). The choice of new elements is weighted according to the labels, favouring elements with a semantic relationship with the last extracted. The strength of the weight is tunable via a parameter η. For $\eta = 1$, this model reduces to the simple PUT model.

In natural systems, the frequency-rank plots feature a variety of system-specific behaviours. The sPUT again reproduces both Heaps' and Zipf's laws, but reproducing other features in detail would require a more detailed modelling scheme than sPUT. For instance, a model could include the distinctions between articles, prepositions, and nouns. Nevertheless, it is interesting that the sPUT model predicts a double slope for Zipf's law as due to the correlations induced by the parameter η (see Supplementary Information of [150] for further details).

One of the main limitations of PUT and sPUT models is that elements introduced earlier will always capture most of the attention. The (s)PUT model does not explain new elements capable of overcoming older ones. Despite the "rich-get-richer" mechanism characterising the dynamics of many natural systems, sometimes new elements become even more successful than the already established ones. To understand how new elements can emerge and diffuse in a population, in [104] is presented a further generalisation of the original PUT model: the Generalised Urn Model with Triggering (GUMT).

The GUMT variant introduces two new key ingredients meant to control the expansion of the *adjacent possible*: (i) an adjustable bias between the choices of retracing the past or looking at the future; (ii) a collective effect on the shape of the space of possibilities. These two ingredients can explain the emergence and evolution of waves of novelties.

The most relevant element is the second. The idea is that not all the elements are equally known to all the individuals exploring the space[8]. When a widespread element occurs, likely all the explorers know it, and there is little bias on the semantic field of the next element. When a niche element appears, the exploration is likely to be confined for a while.

The introduction of these new elements translates into different weights assigned to the elements in the urn. The weights depend on the semantic label κ of the

[8]Here an 'individual' may refer to any constituent of the system that carries its own exploration, being a user of a social network, a collaborator to Wikipedia and so on. In a book, 'individuals' could be scenes or subplots that, having different settings, reinforce and explore different elements.

last element observed and on the element in the urn being known ($\mathcal{A}$) or novel ($\mathcal{B}$). Table 1.2 describes the new weights.

Table 1.2. Weights for the elements in the GUMT model. During the last step was extracted an element with semantic label κ. N_κ is the total number of elements in the urn with the label κ. $N_{\bar{\kappa}}$ is the total number of elements with a different label. These weights are equivalent to those of the sPUT model when $f = g = 1$ and $\eta = \gamma$.

	κ	$\bar{\kappa}$
$\mathcal{A}$	1	$\gamma\ f\left(N_\kappa, N_{\bar{\kappa}}\right)$
$\mathcal{B}$	$g\left(N_\kappa, N_{\bar{\kappa}}\right)$	$\eta\ g\left(N_\kappa, N_{\bar{\kappa}}\right)$

$$\gamma,\ \eta,\ g,\ f \in (0, 1]$$

In real systems at any time can appear an element that reaches an overall frequency close to the maximum (Fig. 1.5 a). In many time intervals, the most frequent element has appeared for the first time recently (Fig. 1.5 b). The sPUT model cannot reproduce this behaviour (Fig. 1.5 c, d) as elements that appeared early are always favourite. By contrast, the weights introduced in the GUMT model to favour novel items allow obtaining results following real cases, Fig. 1.5 (e, f). At any time, newly appeared elements can become the most popular and ultimately reach a significant fraction of the total elements.

1.2 Poisson-Dirichlet Process

We now introduce the two-parameter Poisson-Dirichlet (PD) process, also known as the Pitman-Yor process [114]. This is a well-known stochastic process widely studied in the framework of *non-parametric* Bayesian inference as an extension of the Dirichlet Process. Moreover, the PD process has a marginalisation called the Chinese Restaurant Process, the most used form in inference. For its universality and the links with the PUT model described above, this is the model we will use in the following chapters for the stylometry task.

Non-parametric hierarchical Bayesian models are often based on the Dirichlet or the Poisson-Dirichlet process because their distributions are conjugate with distributions of the multinomial family. In these models, the number of latent variables grows as necessary to fit the data. However, individual variables still follow *parametric* distributions, and even the process controlling the rate of growth of latent variables follows a *parametric* distribution.

The rationale of this approach is clear thinking of a clustering problem. In *parametric* inference, the model would include a parameter for the number of clusters k. We then infer the cluster of each point, e.g. depending on the distance from

its centroid. In cases where we do not know *a priori* the number of clusters (how many literary genres, ethnic groups, hair colours are there?), we have to resort to some criterion like the silhouette score [124] to determine the best value of k. A possible *non-parametric* approach allows the number of clusters to vary following a PD process. However, the probability of a point belonging to a specific cluster will still be parametric (same dependence on the distance from the centroid), and so will be the probability of k via the parameters of the process.

A PD process is a stochastic process whose realisations are probability distributions. Given a metric space $(\mathbb{X}, \mathcal{M})$ and a base distribution P_0 on $\mathbb{X}$, a realisation of the PD process gives a probability distribution P on $\mathbb{X}$:

$$\begin{aligned} P &\sim PD(\alpha, \theta, P_0) \\ P(\cdot) &= \sum_{i=1}^{\infty} p_i \delta_{y_i, \cdot} \end{aligned} \tag{1.14}$$

where the y_i are elements of $\mathbb{X}$ drawn from P_0, and the p_i follow a Poisson-Dirichlet Distribution. The distribution P will be atomic, i.e. discrete, almost surely independent from the nature of P_0. For the moment, we assume P_0 continuous. We will discuss the differences when using a discrete base probability in section 1.2.3.

The probability distribution in equation (1.14) is an *Impulse mixture model*, a class of models that, given a probability distribution P_0 on the measurable space $\mathbb{X}$, yield a distribution over a countable subset of $\mathbb{X}$. The distribution is a weighted sum of δ distributions over points of $\mathbb{X}$. For equation (1.14) to represent a PD process, the y_i points are independently and identically distributed samples from P_0, and the p_i are PD distributed.

Sampling from a PD process (or any Impulse mixture model) provides a partition of the samples.

Definition 1.2.1 (Partition)**.** A partition P of a countable set S is a set of subsets P_i such that $P_i \cap P_j = \varnothing \; \forall i, j$ and $\bigcup_{i=1}^{k} P_i = S$. The partition size $|P|$ is the number of subsets.

In the following, we will call $k := |P|$ and $n_i := |P_i|$.

Let us take a sequence of samples $x_1, \ldots, x_n$ from $P \sim PD(\alpha, \theta, P_0)$. The sampling associates them to k different elements in $\mathbb{X}$ $(x_1^*, \ldots, x_n^*)$ with $\sum_{i=1}^{n} \delta_{x_i^*, y_j} = n_j$, $j = 1, \ldots, k$.

To obtain an infinite-dimensional probability vector $\vec{p}$ with PD distributed components it is possible to follow two steps:

1. build the infinite-dimensional probability vector $\tilde{p}$ following a two-parameters Griffiths-Engen-McCloskey (GEM) distribution;

2. sort the elements in $\tilde{p}$ so that $p_1 \geq p_2 \geq \dots$ and define $\vec{p} := (p_1, p_2, \dots)$.

To build the vector $\tilde{p} \sim \text{GEM}(\alpha, \theta)$ we follow a stick-breaking model:

- take a stick of length 1;
- draw $V_1 \sim Beta(1-\alpha, \theta+\alpha)$;
- break the stick in two parts of length V_1 and $(1-V_1)$;
- call $\tilde{p}_1 := V_1$ and consider the remainder;
- for every $i \geq 2$:
  - draw $V_i \sim Beta(1-\alpha, \theta+i\cdot\alpha)$;
  - break the remainder of the stick into two sections of lengths V_i and $(1-V_i)$;
  - call $\tilde{p}_i = (1-V_1)\dots(1-V_{i-1})V_i$ and continue with the remainder.

A formal definition of the GEM distribution is:

Definition 1.2.2 (GEM distribution)**.** Given parameters $0 \leq \alpha < 1$ and $\theta > -\alpha$, the V_i are independent random variables distributed as $Beta(1-\alpha, \theta+i\cdot\alpha)$ with $i \geq 1$.

The Griffiths-Engen-McCloskey distribution with parameters α and θ, or $\text{GEM}(\alpha, \theta)$, is the distribution of $(p_1, p_2, \dots)$ where:

$$\begin{cases} p_i = V_i, & i = 1 \\ p_i = (1-V_1)\dots(1-V_{i-1})V_i, & i \geq 2 \end{cases}$$

The parameters α and θ are usually termed discount and concentration parameters respectively. It is a custom to call "*concentration*" a parameter that behaves like the inverse of a variance.

1.2.1 Predictive Probability

The PD process is also called a discrete-time discrete-space stochastic process because it can be defined as a sequence of elements $x_1, x_2, \dots$ from $\mathbb{X}$. As a stochastic process, the PD process can provide probabilities for future events. The task is the following: given the first n events in a sequence $x_1, \dots, x_n$ corresponding to k_n different elements $y_j \in \mathbb{X}$, what is the probability that the next event will be $x_{n+1} | x^*_{n+1} = y$ where y can be one of the already seen events y_j or a new one $y \sim P_0$?

The conditional distribution with $\vec{p}$ marginalised out is:

$$P(x^*_{n+1} = \cdot | x_1, \dots, x_n, \alpha, \theta, P_0) = \frac{\theta + k_n\alpha}{\theta+n} P_0(\cdot) + \sum_{j=1}^{k_n} \delta_{y_j,\cdot} \frac{n_j - \alpha}{\theta+n} \tag{1.15}$$

Where:

1. k_n is the number of distinct elements in the first n of the sequence,
2. n_j is the number of elements $x_i | x_i^* = y_j$, with x_i^* the identity of element x_i, and $y_i, i \in [1, k_n]$ element in $\mathbb{X}$ with base probability $P_0(y_i)$. The sum of the multiplicities $\sum_{j=1}^{k} n_j = n$ the number of elements.
3. α and θ are the *discount* and *concentration* parameters of the process, taking values $\alpha \in [0, 1)$, $\theta > -\alpha$.
4. P_0 is the base distribution defined over the space of the tokens $\mathbb{X}$.

From this sequential sampling of the partitions derives the Chinese Restaurant analogy. In this analogy, there is an infinite line of customers (the samples from the PD process) entering a restaurant with infinite tables, each serving a different menu (the samples from P_0). The sampling proceeds as follows:

- the first customer enters and seats at an empty table;
- after n customers have entered, the $n+1$-th customer enters and sees k_n occupied tables with n_i others eating the menu y_i;
- the new customer may sit at an empty table with probability $\frac{\theta + k_n \alpha}{\theta + n}$ picking a new menu y_{k+1} from P_0;
- Alternatively, he can choose one of the already occupied tables i with probability $\frac{n_j - \alpha}{\theta + n}$ and eat menu y_i.

It is possible to use the CR process as an alternative sampler for the PD process where the vector $\vec{p}$ is not known beforehand. In our case, we will be interested in sampling the unknown test using the vector $\vec{p}$ specific to each author. Therefore, we will sample from the correct PD distribution knowing the previous production of the author representing the first n customers seated in the restaurant. This is useful in inference as often one is not interested in the $\vec{p}$ itself. Instead, the $\vec{p}$ is marginalised out, and equation (1.15) requires only the parameters α and θ, the base distribution P_0 and the association between the elements x_i and the classes y_j. This last piece of information is usually one of the goals of the inference.

In the following, when referring to the number and size of the partitions of n independent samples from a PD process, we will use Eq. (1.15) instead of Eq. (1.14) for the ease of mathematical treatment.

It is interesting to note some key behaviours of the probability in Eq. (1.15). First, we note the rich-gets-richer effect induced by the second term of (1.15). The first token comes from P_0, with probability 1. The following ones will have a push towards the elements with a higher number of occurrences n_j. Here we understand the reason for the name *concentration parameter* for θ: the higher the value of θ, the smaller the reinforcement term, the more peaked the distribution of the n_j. The

magnitude of the rich-gets-richer effect depends α. Introducing a new, unobserved token in the sequence increases the value of k. When $\alpha > 0$, this increases the probability of introducing even more new elements. However, a large α also reduces the chances that any single element is reinforced. The reinforced elements quickly become favoured for further reinforcement, giving a skewed distribution[9].

The second behaviour regards the expansion in the *adjacent possible*. Indeed, the first term on the right-hand side of Eq. (1.15) refers to the probability that x_{n+1} takes a value that has never appeared before, i.e. a novel element. A novel element appears with probability $\frac{\theta+k_n\alpha}{\theta+n}$, depending on the total number n of elements seen until time n and the total k_n distinct elements seen until time n. In this way, in the PD process, the concept that the more novelties are actualised, the higher the probability of encountering further novelties is implicit. The second term in Eq. (1.15) weights the probability that x_{n+1} equals one of the previously occurred events and differs from a bare proportionality rule when $\alpha > 0$.

When $\alpha = 0$, there is no expansion in the *adjacent possible*, no triggering of innovations. Such process is called Dirichlet Process, is widely used in inference too and is equivalent to the Hoppe Urn model.

The ubiquity of the PD process is due to its ability to mimic actual data and to the ease of its mathematical manipulation. The PD process is a good model for many systems. It is useful in many contexts as it produces sequences that exhibit all three laws considered in the previous sections. At the same time, it is useful in inference because of the fundamental property of exchangeability. We will now briefly show these features.

Power Law Behaviours

In Eq.(1.15), we observed two key aspects. First, a reinforcement in a rich-gets-richer style, second, positive feedback of introducing a new element on the probability of introducing more novelties.

Heaps' law Let us start with this second aspect looking at how it introduces Heaps' law. To verify that the PD process produces a sub-linear power-law growing number of different elements, we start considering the probability of increasing the number of different elements:

$$\text{Prob}(k_{n+1} = k_n + 1) = \frac{\theta + k\alpha}{\theta + n} \tag{1.16}$$

[9] With $\alpha = 0.8$, an element reinforced once ($n_j = 2$) has a six times higher probability to be selected again than an element with $n_{j'} = 1$. The skewness induced by α is evident in the emergence of power-law tails (see page 22, 'Zipf's law').

If we use a continuous approximation, it becomes

$$\frac{\partial k}{\partial n} = \frac{\theta + k\alpha}{\theta + n} \tag{1.17}$$

With the boundary condition $k(0) = 0$. This can easily be solved by separation of variables, leading to:

$$k(n) \sim \frac{\theta^{1-\alpha}(\theta + n)^{\alpha}}{\alpha} - \frac{\theta}{\alpha} \tag{1.18}$$

The limit exponent in the growth of the number of different tokens, the Heaps' exponent, is exactly α. When $\alpha = 0$, the power-law fails, the growth is only logarithmic and, instead of a Poisson–Dirichlet Process, we recover the simple Dirichlet Process. Note that the Poisson-Dirichlet process is only defined for $\alpha < 1$. It predicts a sub-linear power-law behaviour for $k(n)$ but cannot reproduce a linear growth.

The exact expression for the expected value of $k(n)$, without the continuous approximation, can be found in [23]:

$$\mathbb{E}_{\alpha,\theta,n}[k] = \frac{\theta}{\alpha}\frac{(\theta + \alpha)_n}{(\theta)_n} - \frac{\theta}{\alpha} \tag{1.19}$$

In the limit of $n \gg \theta \gg \alpha$, this expression approximates to

$$\mathbb{E}_{\alpha,\theta,n}[k] \sim \frac{\theta^{1-\alpha}(\theta + n)^{\alpha}}{\alpha} \mathrm{e}^{\frac{\alpha}{2\theta}} - \frac{\theta}{\alpha} \tag{1.20}$$

Taylor's law As a second power-law, we show Taylor's law for the fluctuations. To evaluate the amplitude of the fluctuations in the continuous approximation, we will consider the evolution of a group of processes. At step n_0 every process has $k(n_0) = k_0 + \delta k$ different elements, with $\langle \delta k \rangle = 0$ and $\langle \delta k^2 \rangle = \sigma_0$. Eq. (1.17) with boundary condition $k(n_0) = k_0$ has the solution:

$$k(n) \sim \left(k_0 + \frac{\theta}{\alpha}\right)\left(\frac{\theta + n}{\theta + n_0}\right)^{\alpha} - \frac{\theta}{\alpha} \tag{1.21}$$

Taking the derivative in k_0 we get:

$$\frac{\partial k}{\partial k_0} = \left(\frac{\theta + n}{\theta + n_0}\right)^{\alpha} \tag{1.22}$$

And thus the variance of the group of processes at step n will be

$$\left\langle \delta k^2 \left(\frac{\partial k}{\partial k_0}\right)^2 \right\rangle \propto \sigma_0^2 (\theta + n)^{2\alpha} \tag{1.23}$$

Due to the stochastic nature of the process, at any time, two processes may differ in k. This difference is then amplified through the systems' evolution. In the same limit $n \gg \theta \gg \alpha$, the exact expression for the variance [23] is

$$\mathrm{Var}_{\alpha,\theta,n}[k] \sim \frac{\theta^{1-2\alpha}(\theta + n)^{2\alpha}}{\alpha} \mathrm{e}^{\frac{\alpha}{\theta}} \tag{1.24}$$

Thus the variance grows with a power-law of exponent 2α, and we recover $\sigma \propto \mathbb{E}_{\alpha,\theta,n}[k]$, i.e. Taylor's law.

Zipf's law To derive the Zipf's exponent, we will use the continuum approximation following a master equation approach similar to the one for the PUT model in [149], Appendix B. Let us call N_i the number of elements that occurred exactly i times in the sequence $\{x\}_1^n$. The variation of N_i during the next step will depend on the probability that any of the elements that occurred $i-1$ times is selected again (increasing N_i by 1) and the probability of selecting again an element occurred i times (reducing N_i by 1). In formulas:

$$\frac{\partial N_i}{\partial n} = N_{i-1}\frac{i-1-\alpha}{\theta+n} - N_i\frac{i-\alpha}{\theta+n} = \frac{N_{i-1}(i-1-\alpha) - N_i(i-\alpha)}{\theta+n} \tag{1.25}$$

We can look at the last numerator as (minus) the variation over i of the quantity $(i-\alpha)N_i$, thus writing:

$$\frac{\partial N_i}{\partial n} \approx -\frac{1}{\theta+n}\frac{\partial (i-\alpha)N_i}{\partial i} \tag{1.26}$$

Our goal is to find a law for the fraction f_i of tokens appearing i times in the sequence. We can safely assume that, in the long run, the f_i will tend to some stationary distribution. We can thus write N_i as the product of a fraction f_i independent from time and the total number of different tokens k.

$$f_i\frac{\partial k}{\partial n} = -\frac{1}{\theta+n}\frac{\partial (i-\alpha)f_i}{\partial i}k \tag{1.27}$$

Introducing Eq. (1.17) in the first term, we can simplify the $\theta+n$ and obtain:

$$f_i = -\frac{k}{\theta+k\alpha}\frac{\partial (i-\alpha)f_i}{\partial i} \approx -\frac{1}{\alpha}\frac{\partial (i-\alpha)f_i}{\partial i} \tag{1.28}$$

Where the last step assumes $k\alpha \gg \theta$. This can be solved by separation of variables and yields:

$$f_i \propto (i-\alpha)^{-1-\alpha} \tag{1.29}$$

We obtain the desired power-law behaviour and an exponent for Zipf's law $\frac{1}{\alpha}$, i.e. the inverse of the Heaps' exponent.

As noted in previous sections, this relationship holds only on the tails of the data. Asking for $k\alpha \gg \theta$ is about asking $\alpha n^\alpha \gg \theta$, if α is small, it might be difficult to observe it on data. Note also that α introduces a bias toward small values of i. When $\alpha \to 1$, it is difficult for an element to get reinforced, and there are many elements occurring only once.

Exchangeability

The PD process enjoys the important property of exchangeability. Consider an infinite sequence $X^{(\infty)}$ defined in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The elements of $X^{(\infty)}$ take values in the metric space $(\mathbb{X}, \mathcal{M})$. A sub-sequence of $X^{(\infty)}$ will be $\{x_n\}_{n\geq 1} \in \mathbb{X}$. The sequence is exchangeable if it enjoys the following property:

Definition 1.2.3 (Exchangeability)**.** A sequence $X^{(\infty)}$ is said to be exchangeable if, for any finite sub-sequence of length $n \geq 1$ and any permutation π of the indices $1, 2, \ldots, n$, the probability distribution of the random vector $(x_1, \ldots, x_n)$ coincides with the distribution of $(x_{\pi(1)}, \ldots, x_{\pi(n)})$.

This property is remarkable in inference because of de Finetti's theorem. It states that the only infinite exchangeable sequences are convex combinations (i.e. mixtures) of laws of independent identically-distributed random variables.

To appreciate its relevance for inference, we will state it differently. Consider the probability space $(\mathcal{P}_{\mathbb{X}}, \mathcal{W}, Q)$ of all the measures over $\mathbb{X}$. This means that for every $P \sim Q$, we have a probability space $(\mathbb{X}, \mathcal{M}, P)$. Consider now a set $A = A_1 \times \ldots \times A_n \times \mathbb{X}^{\infty}$ where the A_i are elements of $\mathcal{M}$ and $\mathbb{X}^{\infty} = \mathbb{X} \times \ldots \times \mathbb{X}$. The sequence $X^{(\infty)} \in A$ when $x_i \in A_i$ for $i \leq n$, with no restrictions on the elements x_i for $i > n$.

De Finetti's theorem states that sequence $X^{(\infty)}$ is exchangeable if, and only if, satisfies the following relation:

$$\mathbb{P}\left[X^{(\infty)} \in A\right] = \int_{\mathcal{P}_{\mathbb{X}}} \prod_{i=1}^{n} P(A_i) Q(dP) \tag{1.30}$$

Where the independence is evident from the use of the product, and Q is the *de Finetti measure* of the sequence $X^{(\infty)}$. The de Finetti measure weights all the P participating in the mixture. In other words: conditional on a random probability measure P from Q, $X^{(\infty)}$ is a sequence of independent and identically distributed random elements with a common probability distribution P[10].

Often in inference, we encounter hierarchical mixture models, for example, in the mentioned case of clustering. We can thus reformulate the problem we posed at the beginning of section 1.2. For every data point y_i assigned to a cluster x_i, we need the probability of the point given the cluster $f(\cdot \mid \cdot)$ but also the probability of the cluster itself. We can consistently draw the probability distribution of the

[10] This is easy to see thinking of an exchangeable sequence $X^{(\infty)}$ whose elements take values in $\{0, 1\}$, and a probability distribution Q over the parameter p of a Bernoulli distribution P.

clusters from the *de Finetti measure* that acts as a prior.

$$\begin{aligned} y_i \mid x_i &\overset{\text{iid}}{\sim} f(\cdot \mid x_i) \\ x_i \mid P &\overset{\text{iid}}{\sim} P \\ P &\sim Q \end{aligned} \tag{1.31}$$

To do so, use a PD process for the probabilities of the x_i.

Exchangeability is a powerful property that simplifies the sampling procedure, making it independent of the input order[11]. However, it is also a strong and sometimes unrealistic assumption about the data's lack of correlations and causality.

1.2.2 Conditional Probability

The property of exchangeability may not be evident from Eq. (1.15). However, it stands out when explicitly computing the conditional probability of the whole sequence given the parameters:

$$P(w_1, \dots, w_n \mid \alpha, \theta, P_0) = P(w_n \mid w_1, \dots, w_{n-1}, \alpha, \theta, P_o) \dots P(w_1 \mid \alpha, \theta, P_0) \tag{1.32}$$

For the moment, we consider, as in Eq. (1.15), the case of a continuous base distribution P_0. In this case, each y_j has zero probability of being extracted from P_0 twice. All the elements $x_i \mid x_i^* = y_j$ appeared by reinforcing the first such term appeared. In the case of an atomic P_0, we should consider the number of extractions from P_0 of the same element, see section 1.2.3.

Looking at Eq. (1.15), we notice that the denominator is independent of the identity of x_i and from it being reinforced or drawn from P_0. In the product of Eq. (1.32), we can refactor the contribution of the denominators as a term $\prod_{i=1}^{n} \frac{1}{\theta+i}$.

Let us consider now an element that entered the sequence at step n' and appeared n_j times, i.e. an element $y_j \mid \sum_{i=1}^{n} \delta_{y_j, x_i^*} = n_j$. The weight of this element in the n_j extractions has two contributes:

- a term $(\theta + \alpha k_{n'}) P_0(y_j)$ for the first extraction from the base probability;
- a term $\prod_{i=1}^{n_j - 1} (i - \alpha)$ for the $n_j - 1$ times the element has been reinforced.

The second term does not depend on whatever happened in the sequence. No matter how many steps the process has been running or the number of elements $x_i^* \neq y_j$ appeared.

The first term depends on the sequence through the number of different elements $k_{n'}$ already appeared before y_j. Looking again at Eq. (1.32), we notice that all the k_n different elements in the sequence will bring a term of the same form where $k_{n'}$ will

[11] In the clustering example, considering independent data points, we do not want the result (number and size of the clusters) to depend on the order of the points.

assume all the values in $[0, k_n)$. We can refactor this term as $\prod_{k=0}^{k_n-1}(\theta+\alpha k)\prod_{j=1}^{k_n} P(y_j)$ and the conditional probability finally reads:

$$P(n_1,\dots,n_k \mid \alpha,\theta,P_0) = \frac{(\theta \mid \alpha)_k}{(\theta)_n}\prod_{j=1}^{k}\left[P_0(y_j)\,(1-\alpha)_{n_j-1}\right] \tag{1.33}$$

where $(x)_N$ is the *Pochhammer symbol*: $x(x+1)\dots(x+N-1) = \Gamma(x+N)/\Gamma(x)$ and $(x \mid K)_N$ is the *Pochhammer symbol with increment* K: $x(x+K)\dots(x+(N-1)K)$. The latter is known also as the *Pochhammer k-symbol* and expands as [39]

$$(x \mid K)_N = K^N \times \left(\frac{x}{K}\right)_N = K^N \frac{\Gamma\left(\frac{x}{K}+N\right)}{\Gamma\left(\frac{x}{K}\right)} \tag{1.34}$$

reducing to the *Pochhammer symbol* when $K = 1$.

The probability of Eq. (1.33) is composed of two parts. One is the probability given by the PD process to the partition of n elements in k classes with n_i elements per class $\left(\frac{(\theta|\alpha)_k}{(\theta)_n}\prod_{j=1}^{k}(1-\alpha)_{n_j-1}\right)$. The other is the probability of selecting the associations between each class and the elements of $\mathbb{X}$, i.e. $\left(\prod_{j=1}^{k} P_0(y_j)\right)$.

The exchangeability of the process is now evident. The probability of the sequence, given the parameters, does not depend on the order of the sequence. The only dependence is through the identity and the multiplicity of its elements.

1.2.3 Discrete Base Distribution

In section 1.2.1, we left behind the discrete case for P_0. This might be more suitable to work on words that take values in a discrete space. The main difference with the continuous case lies in the number of times we can extract a particular element y_j from the base probability.

If P_0 is continuous, then (almost surely) two samples drawn from it have distinct values. We used this property above deriving the conditional probability of a sequence. If P_0 is discrete, on the other hand, then $P_0(y) > 0\ \forall y \in \mathbb{X}$, i.e. there is a finite probability of repeating the same extraction. This implies that the partition induced by the process is partially hidden, and Eq. (1.33) does not hold anymore.

An example can clarify what is hidden. Consider the following sequence:

you should better stop and think before you think think

This short sequence gives the partition:

think	you	and	before	better	should	stop
3	2	1	1	1	1	1

Since the base distribution is atomic, we cannot tell how many of the occurrences of 'think' and 'you' came from distinct draws from P_0. Likewise, we have no means to

distinguish which element got reinforced. So the correct version of Eq. (1.33) in this case would be

$$P(x^*_{n+1} = \cdot | x_1, \dots, x_n, \alpha, \theta, P_0) = \frac{\theta + \tilde{k}_n \alpha}{\theta + n} P_0(\cdot) + \sum_{j=1}^{\tilde{k}_n} \delta_{y_j,\cdot} \frac{\tilde{n}_j - \alpha}{\theta + n} \tag{1.35}$$

Where, with $\tilde{k}$, we mean the total number of extractions from P_0 and by $\tilde{n}_j$ the number of steps that reinforced the j-th element from P_0. All the copies of y extracted from P_0 are identical, so we can group all the elements $x_i \mid x^*_i = y_j$, keeping track of the number of extractions of y_j with a new variable t_j. The conditional probability becomes [148]:

$$P(x^*_{n+1} = \cdot | x_1, \dots, x_n, \alpha, \theta, P_0) = \frac{\theta + \alpha \sum_{j=1}^{k} t_j}{\theta + n} P_0(\cdot) + \sum_{j=1}^{k_n} \delta_{y_j,\cdot} \frac{n_j - \alpha t_j}{\theta + n} \tag{1.36}$$

The values of n_j are available from the sequence while the t_j and $\tilde{k} = \sum_{j=1}^{k} t_j$ are not. The t_j are a new latent variable in the model. In the case of continuous P_0, things simplify as $t_j = 1 \; \forall j$ almost surely. In the discrete case, these are new variables to infer.

Although the discrete P_0 would be the natural choice to work on texts, the presence of latent variables implies that the probability of any sequence with some $n_j > 1$ has to be estimated through sampling. Please refer to [23] for a description of how to sample the t_j. We will conduct the stylometry task under the assumption of a continuous base distribution P_0. We will show that, even in this assumption, the expressiveness of the PD process allows us to model the texts under exam closely.

For a complete and in-depth dissertation of the PD process, we refer to excellent reviews in [114, 23, 37].

1.2.4 Equivalence to the PUT Model

The PUT model produces sequences that are not exchangeable. However, it recovers exchangeability in a particular case, corresponding to a slightly different rule (2) definition from page 12. The drawn element s_t is put back in the urn along with ρ additional copies of it iff s_t is not new; in the other case (i.e. when we apply rule (3)), s_t is put back in the urn along with $\tilde{\rho}$ additional copies of it, with $\tilde{\rho} = \rho - (\nu + 1)$.

In this particular case, the PUT model corresponds exactly to the Poisson-Dirichlet process, with $\theta = \frac{N_0}{\rho}$ and $\alpha = \frac{\nu}{\rho}$. The urn acquires the same number of balls at each step, regardless of whether a novelty occurs. This variant makes the generated sequences exchangeable but imposes the constraint $\rho \geq (\nu + 1)$, and thus, in this case, as for the PD process, we cannot recover the linear growth of $k(n)$.

The PUT model is thus a generalisation of the PD process. It allows for a straightforward extension that explicitly considers correlations. In addition, it can

be easily rephrased in terms of walks in a complex space (for instance, a graph), allowing to consider more complex underlying structures for the space of possibilities.

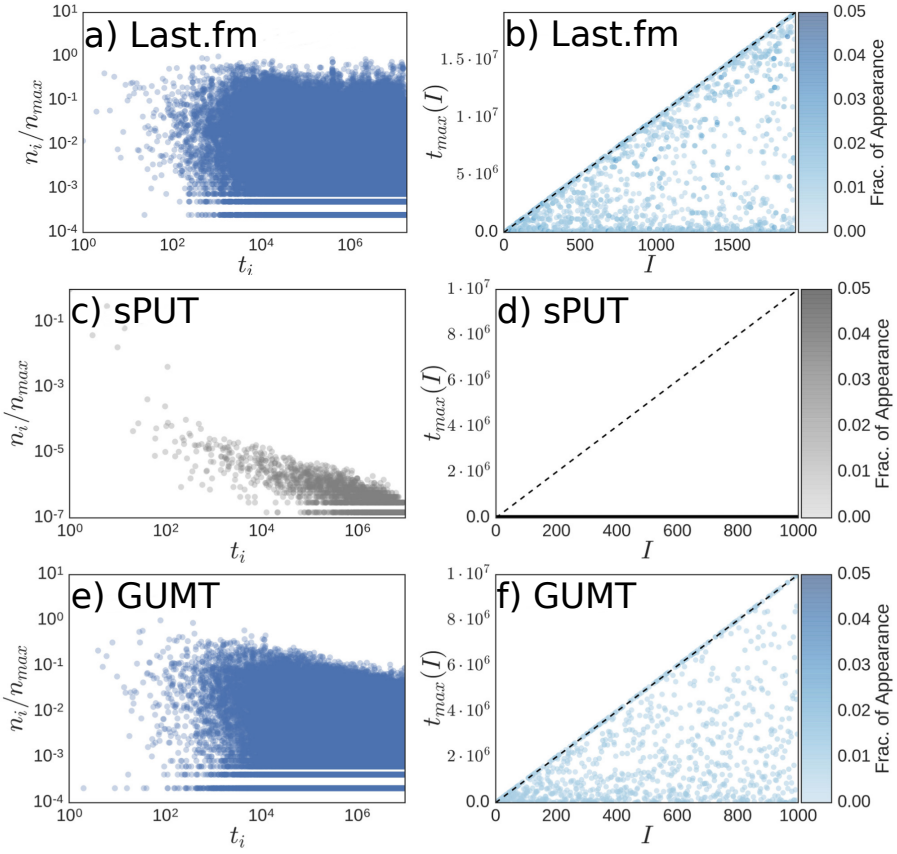


Figure 1.5. Beyond rich-gets-richer. (Left-Panels) Normalised frequency of occurrence n_i of each element in $i \in S$ as a function of its first appearance time t_i. (Right-Panels) For each interval of length $\Delta\tau$ the first appearance time of the most popular element within the interval. Data are shown for the Last.fm dataset (a, b), for the PUT model (c, d) and the GUMT model (e, f). Panels (c, d) are results coming from a simulation of the PUT model with parameters $\rho = 2$, $\nu = 2$ and $\eta = 0.4$ for the model. Panels (e, f) correspond to simulations of the GUMT model with $\rho = 2$, $\nu = 15$, $\eta = 0.001$, $\gamma = 0.004$ and the choice II of the function f and g described in the Materials and Methods section of [104]. Colour is coding for the fraction of time the successful element has appeared within the corresponding interval. The length of the interval is $\Delta\tau = 10000$. Figure from [104].

Chapter 2

Stylometry

This work is part of the large field of stylometry applied, in this case, to authorship attribution. Since its very origins, the techniques of authorship attribution have played an academic and civic role.

The first examples of stylometry appeared with humanism when Lorenzo Valla examined the 'Donation of Constantine' [151]. He based part of his argument on the word choice of the document that was not compatible with other authentic 4^{th} century official texts. In the mid-XIX century, Augustus de Morgan proposed to use word lengths to solve attribution disputes and – by the end of the century – Mendenhall tried to use the statistics of word lengths [102] to distinguish the works of Bacon, Marlowe, and Shakespeare.

In 1954 there was the first known attempt to use stylometry in a trial. Dick Helander, bishop of Strängnäs – Sweden, was convicted for a series of defamatory letters that helped him gain his position two years earlier [71]. Among other evidence, a comparison of frequency lists from the disputed letters with those from his sermons and other documents led to his conviction [6]. However, the expert witness did not use sound statistical analysis, and the case remained dubious.

A few years later, the work of Mosteller and Wallace on the Federalist Papers [106] marked the birth of modern stylometry. The Federalist Papers are a collection of eighty-five essays published in 1787 – 1788 by three different authors (A. Hamilton, J. Madison and J. Jay) to support the ratification of the federalist constitution of the United States. Mosteller and Wallace applied linear discriminant analysis and Bayesian inference to the twelve essays of debated attribution. They attributed eleven papers to Madison, with the last being probably by Madison too.

This chapter is a general overview of the research in stylometry. The reader will find some sections more related to the primary subject of this thesis. In particular, the specific subtask is described in section 2.1.3, and the kind of features used are introduced in section 2.2.2 ('Lexical'). Some related methods we used as a comparison

are presented in sections 2.3.3 and 2.3.4. Our approach belongs to the broad class of *profile-based* approaches. A description of their features is in section 2.4.2.

2.1 Stylometry and its Subtasks

Challenges in stylometry evolved in the nearly sixty years since its birth. Different subtasks in stylometry require different approaches and techniques. However, even if tools and procedures may differ completely, all the main subtasks defined in the past years can be placed on a single scale depending on the number of candidate authors, see fig.2.1 and [80].

With no candidate author at all, we cannot even think of attribution. The best we can do in this case is to attempt *author profiling*. In this case, we try to infer as much information as possible from the given anonymous text. This information may be about the demography, the psychology, or the author's interests.

When there is only one candidate author, we are asked to verify its authorship. The author *verification* task is tricky as we have to tell if the style of the unknown author is close 'enough' to the candidate author. Varying the setting, defining how much is enough may not be easy.

With few candidate authors, we are in the classical setting of *authorship attribution*. In this case, the set of candidate authors is closed and (relatively) small. We have examples of the writing from all the candidates, and we know one of them is the actual author. This is the setting for many literary disputes as to the mentioned case of the Federalist Papers.

With many candidate authors, in the order of the thousands or tens of thousands, we face the *needle-in-a-haystack* problem. Nevertheless, it is still a case of proper *authorship attribution* as we know (or hypothesise) that the actual author is among the candidates. However, the number of candidates makes many *authorship attribution* approaches useless in this setting.

Other interesting subtasks of stylometry are *stylochronometry* and *adversarial stylometry*. In the first case, we are interested in the time ordering of texts. We must place a text with no date in the correct order among others, with or without authorship information. The second case is strongly connected to *authorship attribution* as is

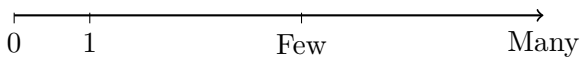


Figure 2.1. Number of candidate authors. This number defines the kind of stylometry subtask involved. For a growing number of candidate authors, we have *profiling*, *verification*, classical *authorship attribution*, *needle-in-a-haystack* problem.

the active attempt to avoid the correct attribution. To this end, typical options are *obfuscation* of the author's style or *imitation* of someone else's style.

All these tasks tackle different sides of the same problem. They seek to unveil the unique style of each author to describe it (*profiling*), to use it for identification (*verification* and *attribution*), to conceal or mimic it (*adversarial stylometry*) or to capture its changes over time (*stylochronometry*). This fundamental similarity implies that the approaches used to address each task share many features. We will briefly outline the main specificities.

2.1.1 Author Profiling

In *author profiling*, also called *author characterisation*, the intent is to describe the potential author [122]. This description may include demographic aspects as age and gender [112, 118], or a psychological characterisation of sentiment or personality [152, 154].

This subtask has several possible applications ranging from forensic to marketing. In forensic analysis, *author profiling* may help assemble a set of suspects for further investigation. In marketing, *author profiling* helps uncover the demographics of customers reviewing products [116]. These reviews often do not disclose data about the author, and *author profiling* can inform business experts for decisions about marketing and business strategy.

The 2019 edition of the PAN workshop [119] presented another application of *author profiling*. In that edition, participants had to extract some particular information: if the poster of a tweet is a bot or a human and, in this case, if it is male or female. This kind of analysis may have a significant impact as Twitter bots had a role in major political events such as the U.S. presidential elections [17] or the referendum for the independence of Catalunya [145]. Moreover, bots may alter a product's public perception by artificially inflating or undermining its popularity.

2.1.2 Authorship Verification

The task of *authorship verification*, also in the forms of *similarity* or *plagiarism detection*, aims to determine if a text has the same author as other texts written by the same pen. Depending on which (alleged) authors' identities are known, this task changes the flavour. If the author of the reference corpus is known and is supposed to be the same author of the anonymous text, it is *authorship verification*. When a different author claims paternity over the anonymous text, it is *plagiarism detection*. When the author of the reference corpus is not known, we are just trying to establish common authorship through *similarity detection*.

This problem is challenging as we usually have samples only from the candidate author. There are two main approaches in this case: using information only from the candidate author [4, 109] or the artificial inclusion of other candidats [83].

When using samples only from the candidate author, the approach requires a threshold to determine if the text is close enough to the candidate author. If using a distance or similarity measure, this threshold is explicitly set. The threshold may also be implicit in a single class classifier trained only with positive samples. The boundaries of the class may be more (less) tight, allowing similar (diverse) samples in the *same-author* class.

The artificial inclusion of other authors, or impostors, transforms this problem into a common *authorship attribution* task. This method, however, has its flaws too. First, there is the problem of the selection of the impostors. The impostors must be chosen to be as similar as possible to the candidate author to present a true challenge. This similarity may include every possible aspect: demographics, favourite topics and of course "style". To gather the perfect set of impostors, we should know how to characterise the author so well that the problem becomes trivial.

As a consequence, we have that the outputs are intrinsically unreliable. When the text is classified as *other-author*, we can trust that it is indeed far from the style of our candidate. When the text is classified as *same-author*, we can only say that none of the impostors was closer than the candidate. There is no general way to determine if this is due to our fault in selecting the impostors, or if the text is indeed from the same author.

A particularly challenging version of this subtask is the *author identification* across different genres [68, 142]. For example, is the author of these blog posts the same as this harassing letter? Is the author of these business letters the same as this suicide note? Solving similar problems may have a great practical impact.

2.1.3 Authorship Attribution

The *authorship attribution* subtask is the earliest and most investigated in stylometry. The first modern attempt in computer-assisted stylometry is the authorship attribution work by Mosteller and Wallace on the Federalists papers [106].

The classical setting of this task includes a few authors [19, 69, 76, 80, 106], usually with many examples of each author's style. In this approach, the best corpus for testing is composed of texts on the same topic written by authors with similar demographics and backgrounds. The idea is to get texts whose main difference is the author itself. The rationale behind this approach is that we cannot rely on gender, age, topic or social differences between the actual author and the alternatives in

any hypothetical use case. The idea behind this kind of corpora is of creating a worst-case scenario where no other hints are available but the very style.

This classical setting ignores many challenges offered by more realistic use cases. In many applications, we have to face imbalances in the available size of the training corpus for the different candidate authors. The topic of the training texts may vary from author to author, within the same author and, ultimately, be different from the topic of the anonymous documents. Moreover, texts may be coauthored and cite or copy other authors' texts. Finding and segregating can be tedious for quotations and challenging for coauthored text.

Recent studies focus on more extensive and less balanced corpora. Since the beginning of the century, corpora grew to hundreds of authors and hundreds of thousands of texts [78] then tens of thousands of authors [131, 81]. Often classical, balanced corpora are still used for benchmark, but no method can call itself *state of the art* without good performance on larger large imbalanced corpora [135].

A different nuance of *authorship attribution* is *author change detection*. This kind of analysis is performed on (supposed) multi-authored texts. In this case, every part of the text, usually a paragraph, must be assigned to the proper author. This kind of challenge has been one of the PAN tasks of the past five editions [160].

2.1.4 Stylochronometry

Ordering texts using stylistic markers date as back as the 19^{th} century. The first recorded attempt [144] is the work of Campbell on the Platonic dialogues [26] of 1867. However, a subjective choice of features characterised the early studies and led to dubious results. More systematic computer-based approaches arrived in the eighties [20, 66] while the term *stylochronometry* was coined only in the late nineties [52].

From an individual [27], a collective [28, 67] point of view, or both [77], language changes over time. This change can include variations on many levels: choices in the vocabulary [66] (i.e. the use of words of Celtic origin as opposed to words coming from Latin), trends in the average word length [28] or more complex dependencies from many predictors [77].

Some works tackle this problem using topic models to track the evolution of the authors' interests [156]. For example, this technique allowed the author to track the interests of scientists over the different editions of the NIPS *Conference and Workshop on Neural Information Processing Systems*.

2.1.5 Adversarial Stylometry

The purpose of *adversarial stylometry* is to deceive stylometry techniques. Deceiving is possible by masking the true identity of an author or making the style of some text resemble some other author's. This branch emerged in early 2000 as a reaction to the progress in other subtasks of stylometry [121]. Narayanan and collaborators [107] showed the "feasibility of internet-scale author identification" only a decade later. Scholars and IT security experts recognise that the new techniques allow connecting the user behind different pseudonyms solely from the content of their messages even in a space crowded as the web.

The approaches to *adversarial stylometry* vary in technique and degree of automation. They range from the almost manual edit of texts [70] to automatic machine translation [21], from neural network approaches [47] to automatic synonym substitution [123] and genetic algorithms [94]. This varied development also led to a number of tools for public use like Anonimouth [99], AuthorCAAT-I [36]/-II [35]/-III [49] and Mutant-X [94].

Despite the developments, one big problem seems to be still far from the solution. Existing automated authorship obfuscation approaches find it challenging to successfully evade machine-learning-based authorship attribution classifiers while preserving the semantics of the text [94]. As a result, the text is (possibly) obfuscated, but it loses its meaning. Sometimes it is completely subverted.

Texts for which the author may seek anonymity are usually meaningful. The challenges in 'preserving semantics' that often seems to be a secondary problem behind obfuscation itself are central. Unfortunately, studies about the effect on obfuscation of the re-establishing of the correct meaning are still lacking.

In the past two decades, the field of adversarial stylometry has been active in many directions. For example, there are now methods to deanonimise executable code of computer programmes [53, 25] and studies to obfuscate their style [25]. Other studies again investigated the possibility to discover if a text has been obfuscated [3].

Ethical Implications

The interest in *adversarial stylometry* is at least twofold. On one side, learning to fake an author's style requires a better understanding of what determines the style itself. On the other side, it can help to mitigate stylometry inherent treats.

Authorship attribution techniques should be indeed regarded as potentially dangerous technologies. Imagine the police of an authoritarian state adopting stylometric methods. Imagine an activist advocating for the rights of their people on a blog or social network, carefully obfuscating their identity, IP address and so on. Imagine

they also write (or wrote) elsewhere innocent posts using their real name. The police could quickly get their name or at least a shortlist of people to investigate further.

As an example of the current concerns about stylometry, consider recent results presented at the NeurIPS-2021 conference. McIlroy-Young and collaborators [100] applied stylometry to chess plays and identified a player among thousands with 86% accuracy. Notably, the paper contains an "Ethical considerations" section introduced as the NeurIPS organisers accepted the paper 'on the condition that the researchers elaborate on the privacy risks' [63]. In the 'Privacy concerns' paragraph, the authors note the similarity with *authorship attribution* in texts and propose approaches to obfuscate the playing style.

A technique devised to help solve academic controversies or help fight unlawful behaviours on the web (stalking, hate speech) may become a threat to the freedom of speech. At the same time, style obfuscation techniques that may help activists may also be used in blackmail, as once done cutting letters from newspapers.

Some stylometry works tackle security issues but doing so pose serious privacy concerns. In their 2015 paper [109], Nirmal and collaborators state:

> If there exists an email server (or an additional security interface) that has sufficient knowledge on the author's writing style; it could easily detect a discrepancy in the email style and could prevent impersonation.

Indeed, this might be an effective security measure against the malicious use of infected email accounts.

Aren't then *adversarial stylometry* techniques allowing criminals to do bad things? Useful approaches, similar to the one investigated in [109], in the wrong hands, may lead to privacy nightmares when used in mass surveillance. Quoting the 'Abuse FAQs' from the Tor Browser [41] support pages[1]:

> Criminals can already do bad things. Since they're willing to break laws, they already have lots of options available that provide better privacy than Tor provides. They can steal cell phones, use them, and throw them in a ditch; they can crack into computers in Korea or Brazil and use them to launch abusive activities [...] Normal people, on the other hand, don't have the time or money to spend figuring out how to get privacy online.

With the adequate changes, this reasoning also applies to privacy and anonymity protected via stylometry techniques. For malicious individuals or organisations is easy to infect many email addresses and try until one of the forged messages passes

[1] https://support.torproject.org/abuse/#what-about-criminals, URL checked on December 30, 2021.

through the filter. Even a single tracked email might be fatal for the anonymous source of some exposed mail to a journalist. The role of whistleblowers in recent years, and the often harsh reaction from governments and organisations, make this topic relevant for researchers and the general public.

In Chapter 9, we investigate the potential threats posed by the method described in this thesis. We conclude that obfuscation tools should be readily available and effective as other privacy tools.

2.2 Data Preparation and Feature Extraction

The texts collected for analysis need preparation before being fed to a stylometric algorithm. Except for very few cases where data are generated specifically for experimental purposes, data used in stylometry come from sources with very different aims. This variety implies that they contain elements as page numbers or HTML tags. After removing or normalising all the superfluous or spurious information, the stream of characters needs further processing to become useful for the analysis. This phase is called *feature extraction* and is strongly dependent on the specific approach.

2.2.1 Preprocessing

The first phase of cleaning depends on the origin of the texts. Books and essays usually come from digital publications or the digitisation of older editions. These include metadata, information about the publisher or the print shop, page and chapter numbers, and any other addition to the text (images, editor's notes, highlights, ...) included to meet the readers' needs or tastes.

More informal kinds of texts are usually a product of digital interactions. Email, weblog, SMS are all valuable material for stylometric analysis. These texts are usually scraped from the web or retrieved from databases. Even in this case, there is plenty of metadata, and the texts contain URLs, tags, escaped sequences (e.g. "&" in HTML documents to represent "&") and so on.

After this first cleanup, there is a second cleaning phase more dependent on the approach. In many cases all letters in the texts are lower-cased. Some approaches need the removal of all the punctuation and other non-alphabetical characters. Finally, in some cases is useful to normalise some kinds of characters either for the needs of the algorithm or because bearing confusing information, not from the author. This step may include unifying the quotes style (changing from "smart" to "plain"), normalising white space (space, non-breaking space, tabulation) and other context or language-dependent passages.

2.2.2 Feature Extraction

After preprocessing the texts, another crucial step is missing before beginning to work on the specific stylometry task. The text is still a continuous stream of characters at this step with no distinction of its constituents. The *feature extraction* process elaborates the texts to produce a set of measurable features that allow comparison.

The features considered in stylometry works span all the possible levels. At the lowermost level, we have *lexical* features that look at the words or even at the bare characters of the text without any additional information. At a higher level, we find *syntactic* features. Here the role of each part of the text is analysed at a syntactical level identifying nouns, complements, verbs and so on. Then, growing in abstraction, we find *semantic* features. Here the very meaning is considered, for example, the use or availability of synonyms. Finally, at the topmost level, *structural* features consider the whole document identifying the different sections and their role. A last class of features – commonly referred ad *domain-* or *application-specific* – may belong to any of these classes depending on the context.

Lexical

Lexical features look at the symbols in the text without trying to infer any additional information. This is convenient as it makes these features largely language-independent.

The most basic *lexical* features are character N-grams, i.e. groups of N consecutive characters from the text stream. Character N-grams are convenient as they find application to most languages, from alphabetic writings as English to logographic as Chinese. Usually, all possible N-grams are considered processing the text with a sliding window of width N.

The second class of *lexical* features are word tokens. These are not language-independent as the language must have word boundaries. In agglutinative languages, a whole, complex sentence may consist of one or a few words. In some cases, punctuation marks are considered tokens on their own.

Misspelling can affect the use of *lexical* features. For example, the word (or trigram) ‘rwd’, as a misspelling for ‘red’, introduces a new, different word token (or trigram, since this sequence of letters is absent in regular English). In some cases, this can be useful for analysing the author’s most common mistakes. However, typos introduce noise when not analysing mistakes and must be removed using some language-dependent orthographic corrector. Furthermore, the corrector may itself introduce noise: in the example before – without knowing the context – ‘red’,

'rod' and 'rad' are all possible corrections[2]. Similar problems are more relevant in an informal context with the use of slang or web jargon as "31337 5p34k" [eleet speak, for 'elite speak', the language of the elite, most skilled users] that modifies the spelling, often substituting letters with numbers with a similar shape.

From these features, it is possible to extract relevant measures. Word length distributions or vocabulary richness measures were mostly used in the early phases of stylometry. Vocabulary richness measures date as back as the forties with Yule's K measure [158]. Modern approaches prefer to use the so-called *bag-of-words*. All lexical features are considered without account for their order in the text and context, focusing on their frequency.

Syntactic

Syntactic features look at patterns in sentence elements. A good set of tools is needed to extract this information: tokenisers, parsers and *part-of-speech* (PoS) taggers. These are not available or equally proficient for every language. The result of the analysis is limited to their quality. When the performance is insufficient, these tools add noise to the extracted features[3].

The tagged text allows the syntactic analysis of the style. This analysis is possible by analysing the tags' frequency or the rewrite rules representing how an author structures sentences. Sidorov and collaborators [137] obtained consistent improvements using dependency trees to build 'syntactic N-grams'. They derived token N-grams measuring token adjacency on dependency trees, i.e. the next token is syntactically dependent on the former.

Semantic

The *semantic* features exploit the very meaning of the words in the text. While approaches like topic modelling 2.3.4 try to infer the relations between the word from their use[4], extracting semantic features relies on specific tools. One of the most commonly used is the WordNet®[5] database of synonyms [103]. Only a few languages have this kind of tool readily available.

An example of use comes from [31], where the authors weighted the word choices with the number of synonyms available on WordNet®. An author sharing many words with the unknown text will receive a lower score if those words have few synonyms. This is because there are few options to express the same concepts using

[2] Also, on English keyboards, 'a' and 'e' are both neighbours to 'w'.

[3] Consider a rough PoS tagger failing to distinguish the verb 'to book' from the object made of paper.

[4] For example, they learn to distinguish the uses of 'bank' next to 'river' or 'loan'.

[5] https://wordnet.princeton.edu/, last checked January 22, 2022.

different terms: it is difficult to avoid using the word 'computer' when writing about computers. On the other end, an author that shares fewer words may get a higher score if they pick those words from many synonyms. For example, few people would use 'verdant' to describe a green area or landscape, which might be a clue.

Structural

Structural features are pertinent to the organisation and the layout of the text. The use of indentation, blank lines to separate paragraphs, greetings and signatures are all examples of *structural* features that characterise an author's style.

Some of these features are generic and apply from essays to blog posts to mail. Others – like the use of a closing signature in emails – are more specific to some domains. In some domains the use of structural features is limited. For example, the layout of published books or articles does not depend on the author but the publisher.

Domain-specific

Among *domain-specific* features, all sorts of the former are possible. Due to the specific knowledge of the task, *domain-specific* features may include some *lexical* features that would be excluded in the standard analysis. Some specific *syntactic* structures may reveal non-native writers, and some keywords may receive additional weight in *semantic* analysis. Even if using an approach that requires removing stop words, words as '`the`', '`who`', '`take`', '`that`' may require a special treatment if studying music reviews.

Hashtags and mentions in Twitter posts are an example of *domain-specific lexical* features. Their use and content may reveal a lot about the author but are not applicable or relevant in other contexts.

2.2.3 Feature Selection

With such a wide offer of features, even approaches that rely only on a few categories often need *feature selection*. Selecting means retaining only part of them and discarding the others. Despite being so relevant for the outcome of the task, the researcher's experience is often the sole guidance in this step.

In stylometry, feature selection indicates a form of dimensionality reduction in data. Often, every different token, N-gram, rewrite rule is a dimension. Documents take place in this space depending on the frequency of each token. A common approach is to ignore the dimensions where many documents have a zero or all documents have similar values.

Every classification approach hints at a general order of magnitude, and some also suggest a rule for the selection. However, over the years, any number of features has been suggested, from the few tens to the many thousands. Also, the rules for selecting the features vary wildly, from corpus dependent inferred relevance to precompiled lists of words specific for language and application.

When inferring the most relevant feature from the corpus, varying the stylometry approach, all sorts of rules have been proposed. For example, the suggestion may be to select only the most common features (appearing in all texts, top N), the least common, those whose frequency varies a lot or a little, or the most informative, only to name a few.

Studies on the best feature sets [42], as well as on the composition of training and test sets [44] or the length of the samples [43], are ongoing as every new approach changes the rules and moves the thresholds.

2.3 Classification Approaches

Over the years, researchers proposed many different approaches to stylometry and its subtasks. Some approaches are very context-specific, others more general. At the same time, given the same task and the same corpus, many different approaches may prove valid.

The following sections offer an overview of the different options following a classification derived mainly from [46, 108].

2.3.1 Machine-Learning

This class of methods reaches extremely high performance through the careful selection of many components. The texts are described as vectors in a multidimensional space. Vectors are then classified or grouped into categories. Here too, the choice of classifiers or clustering algorithms and their parameters is almost unlimited.

Machine-learning classifiers optimise boundaries between predetermined classes. Good boundaries minimise some loss function specific for the algorithm or task. The general purpose of loss functions is to introduce a penalty for misclassified samples in the training set. Loss functions may include penalties for samples too close to the boundaries or different weights for different classes[6].

Machine-learning algorithms are widely used in all stylometry subtasks. For example, during the 2013 edition of the PAN Competition [120], all participants

[6]This tuning can account for the different effects of misclassification or compensate for the class imbalance. For the first case, think of the different weights of a false negative or a false positive in a court trial.

used a machine-learning algorithm for classification, including decision trees, support vector machines, logistic regression, and random forests.

Clustering algorithms are an option when the interest is in discovering relationships within the corpus. For example, a clustering algorithm can find stylistic similarities or reduce the search space for authorship attribution problems [64]. Clustering is an unsupervised machine-learning procedure. The algorithm derives a natural separation of the feature space that may or may not correlate with the class labels. While in classification, class labels are known and incorporated in the classification process, in clustering, they might be absent altogether, and there is no guarantee – nor specific aim – that all the texts from the same class (author, gender, age) will fall in the same cluster.

Support vector machines (SVMs) have been a common classifier choice in use since early 2000 [40] and included in stylometry software as Stylo [45]. SVMs can handle large and sparse datasets with large combinations of features of different nature. The examples of SVM classifiers, often compared to other machine-learning approaches, are countless. As an example, in [1], an SVM is compared to decision trees while, in [58], the comparison is with Naïve Bayes, K-Nearest-Neighbours (KNN) and decision trees. SVMs are usually among the best but performance varies wildly across studies. An interesting case is [109], where the authors train a single-class classifier to recognise an individual's email style.

Artificial Neural Networks (ANN) have been used in stylometry since the nineties [76] and reached very high performance. For example, the approach in [13] obtained the best overall result in the Author Identification task at PAN2015 [143]. However, ANNs require lots of training data [54], and the limits to explainability in their results [117] may limit their range of application. On the other hand, it is relatively easy to build good language models using ANN and to use them to generate obfuscated text in *adversarial stylometry* as in [136].

2.3.2 Similarity

Similarity approaches measure the similarity and differences between two texts to determine if the same person wrote them. Similarity approaches are also often referred to as *inter-textual distances* even if their output is not always a distance in the sense of a mathematical function, i.e. being positive semi-definite, symmetric and obeying triangle inequality. These methods often rely on the vocabulary used in the two texts, measuring some (weighted) degree of overlap.

The main idea of similarity approaches is that if the elements used in two texts are similar, the texts are close and may be written by the same person. The result "these two texts are closer" is obtained with specifically crafted distance measures.

KNN classification algorithms use a similarity approach when K=1. We do not include similarity approaches in the machine learning group as generally denoted by the simplicity of their attribution rule (i.e. first nearest neighbour), their general inability to deal with sparse data and the care in selecting the distance function rather than Euclidean distance in $\mathbb{R}^n$.

Several distance metrics have been created or adopted for this purpose, including Burrows' Delta [24], Chi-Square [130], Kullback-Leibler Divergence [12, 130, 161], Stamatatos distances [140], Argamon's Delta [10], Eder's Delta [125] and others. The Burrows' Delta is simply the difference in z-scores of the relative frequencies of the most frequent words in texts. Its alternative versions include weights depending on the word rank (Eder), the square of the relative frequencies (Argamon) and so on.

Compression-based methods [16, 86] represent a different approach to similarity measures. These methods also minimise some distance between train and test samples. First, all texts from an author are joined to produce a file A. A compression algorithm produces the compressed file $C(A)$. Next, the unknown text B is added to A and compressed to obtain $C(A+B)$. Finally, the difference in bits between $C(A)$ and $C(A+B)$ measures the similarity (actually a function of the crossentropy).

A known [96] drawback to compression-based approaches is slow running time: the corpus of every author has to be compressed again for each unknown text. The authors of [96] also note that compression models are easy to apply and require little to no preprocessing. These methods continue to yield state-of-the results for *authorship attribution* [110, 108].

2.3.3 Probabilistic

Probabilistic approaches apply the Bayes rule to get an estimate of the probability that some author $\mathcal{A}$[7] produced some text $\mathcal{B}$:

$$P(\mathcal{A} \mid \mathcal{B}) = \frac{P(\mathcal{B} \mid \mathcal{A})P(\mathcal{A})}{P(\mathcal{B})} \tag{2.1}$$

Then an author $\mathcal{A}^*$ is proposed following a *maximum a posteriori* (MAP) principle $\mathcal{A}^* = \arg\max_{\mathcal{A}}(P(\mathcal{A} \mid \mathcal{B}))$. Being $P(\mathcal{B})$ a normalisation constant, key elements to determine the probability are $P(\mathcal{A})$ and $P(\mathcal{B} \mid \mathcal{A})$.

The prior probability is usually taken uniform or derived from the data. In the first case, we may ignore some relevant information but might be a safer assumption in case of imbalanced training data. The results with uniform priors will be the same as through *maximum likelihood estimation* (MLE).

[7] Here, by A, we mean an individual's identity or a label corresponding to some profile.

Most of the researchers' effort is usually in determining the conditional probability $P(\mathcal{B} \mid \mathcal{A})$ (at least since Mosteller and Wallace [106]). In a Naïve Bayes approach, all the features are assumed to be independent. We are not interested in the exact probabilities but only in the identity of the most probable author. Even if independence may seem a rough assumption, this approach often leads to good results [58]. Researchers proposed several options to overcome the limitations of the assumed independence. In [76], the authors introduce the covariance matrix, but this is feasible only with non-sparse data. A different approach is to include a language model that assigns probabilities to sequences of words. This is the case of [113], where the authors develop a 'Chain Augmented Naive Bayes' using a token N-gram model similar to those used in automatic speech recognition [79].

2.3.4 Topic Modelling

Topic modelling techniques were developed in the late nineties for dimensionality reduction in information retrieval. Topic models assign every word in a document to a topic trying to resolve polysemy issues. The document is first represented as a vector of word frequencies. After assigning words to topics, the document representation is a mixture of topics. This achieves the result of projecting the document from the N-dimensional space of the words, N in the thousands or hundreds of thousands, to an M-dimensional space of topics, M usually in the hundreds.

Topic models have found their application in *authorship attribution* for at least ten years [134, 157]. Their fortune is due to the possibility to specialise their generative model to capture the relevant information.

In [108], topic models are considered a kind of feature. Indeed, these approaches try to extract semantic information from lexical features. However, there is a relevant difference that suggests a different classification. Methods using syntactic or semantic features usually rely on external tools for their extraction and then use these features for further processing using any of the approaches described. In topic modelling, the characterisation of the relations between tokens is central to the analysis. The output then undergoes little other processing, with the last stage being probabilistic or similarity-based.

One of the most relevant and influential topic model implementations is the Latent Dirichlet Allocation (LDA) [18]. LDA is a three-level Bayesian technique for modelling a collection over a set of topics. In its base definition, LDA models documents as a mixture of topics where each topic is represented by a multinomial distribution over words. The mixture weights of the topics for each document and of the words for each topic are drawn from a Dirichlet distribution. The relevant information is in the estimate of the mixture weights. The weights in each topic's

word distribution give an idea of its semantic field. At the same time, the weights in the topic distribution point out the most relevant topics for each document.

LDA will not give direct information on authorship. However it is possible to compare the topic distribution of known authors with the one estimated on the unknown text. For example, Hellinger distance [134] between the topic distributions, even in this simple approach, offers interesting results close to those obtained using SVMs.

Seroussi and collaborators introduced [133] the Disjoint Author-Document Topic (DADT) Model. This model aims to "separate document words from author words by generating them from two disjoint topic sets". This approach incorporates the intuition that the document itself dictates some words while others are introduced as specific to the author's style. Thus, there are two kinds of topics in the DADT model: *document* and *author topics*[8]. At the word level, the generation follows the following rules: (i) choose to draw a word from *document* or *author topics* with a document-specific probability; (ii) if *document topics* are selected, choose a topic from the document's *document topic* distribution; (iii) if *author topics* are selected, choose from the *author topic* distribution of one of the document's authors; (iv) choose a word from the selected topic. In this case – given the model's parameters, the test text words, and the inferred *author/document topic* ratio and document topic distribution – the probability of each author is determined directly.

For the subject of this thesis, it is relevant to mention an extension of the DADT model that the authors presented as "Probabilistic DADT" or DADT-P [135]. This model gave the best result on almost all the corpora the authors applied it to. Only in a few cases it was slightly ($< 2\%$) outperformed by an SVM. In the DADT-P model, every candidate author $\mathcal{A}$, in turn, is considered to be the actual author of the unknown text $\mathcal{B}$. The candidate authors have their topic distributions fixed after the training phase. Each topic's distribution over words is fixed too. The authors infer the text's *document topic* distribution and the *author/document topic* ratio and compute the probability of obtaining the unknown text from the candidate author $P(\mathcal{B} \mid \mathcal{A})$. In [135], the authors infer the distribution of documents over authors, thus obtaining a proper $P(\mathcal{A} \mid \mathcal{B})$ via the Bayes rule. This choice is delicate as it might be strongly affected by class imbalance if the test set is instead balanced, as suggested in [142] for *authorship attribution* tasks.

[8] Please note that the term 'topic' in 'topic model' does not match exactly its usual meaning. Some topics in topic modelling may include mostly stop words and be mainly topic-independent (in the ordinary meaning of this expression).

2.3.5 Complex Networks

Stylometry did not escape the attention of scholars in complex network theory. A common approach to building a network from a document is removing the stop words and then considering the remaining words as nodes. Then (directed) links are added from every word to the following in the text. Links can be weighted by the number of times a pair appears in the text. Several studies were devoted to the study of the properties of such networks and their application to stylometry [88, 101], with some interesting findings as a power-law scaling in the number of observed links (word bigrams) as a function of the number of different words [101].

When using the networks for stylometry, the approaches are pretty diverse. The authors of [9] based their analysis on global properties of the network, such as clustering coefficient or degree correlation. They treated these as features for classification of the texts. In [109], when building the network, words are connected if appearing within a distance n and stop words are not removed. The links of highest weight and the nodes of highest total degree are fed to an SVM as features. Amancio [7] obtained some results from the network variation over a book's development. The author applied machine learning techniques to the Fourier coefficients of the series of values for clustering, average shortest path, betweenness and accessibility.

De Arruda and collaborators showed in [50] a different approach that they defined 'mesoscopic'. In their approach, the nodes are groups of paragraphs, and the weight of the link between two nodes depends on the overlap of the most relevant terms for the two extracts. This network representation was later [95] applied to authorship attribution feeding some network statistics (degree, assortativity, centrality) to machine learning classifiers. This mesoscopic approach distinguishes natural and synthetic texts better than the co-occurrence networks. However, it is not so proficient in finding the author of a text despite many text samples used. However, the representations of the evolution of the books [95, 8] are truly picturesque.

2.3.6 Meta-Learning

The last class of approaches includes meta-learning models. Here one or more of the approaches mentioned above are used not to get an immediate result but some new feature for further analysis. For example, in [84], a semi-supervised method is presented. First, texts are classified using an SVM. Then, if a Common N-gram classifier confirms its prediction, the documents are added to the training set to improve the results on the remaining documents.

Koppel proposed in [82] an interesting *authorship verification* approach based on the performance of an SVM. First, the classifier is trained to distinguish the

chunks of the text under exam from other texts of a single author. Then, the k most relevant features are removed and the classifier trained again. The procedure continues observing the degrading performance of the classifier. If the author is the actual author, the style will be similar, and only a few features will allow the distinction of the text. After only a few iterations, the classifier will lose most of its power.

Kusakci proposed [87] a 'committee' of neural networks. Since every network may vary in authorship attribution power, the confidence on every machine is evaluated from its accuracy over training samples similar to the test sample in consideration. Hence, all the machines cast their votes weighted with their accuracy.

2.4 Instance- and Profile-Based Approaches

The above classification offers an overview of the available options. However, some essential features that characterise every approach may have passed unnoticed.

Here we recover the distinction between *instance-based* and *profile-based* approaches. This distinction was considered one of the most fundamental properties of the attribution methods by Stamatatos in his seminal 2009 review [142]. This relevance is undoubted for the philosophy behind each method and may limit the fields of application of each approach.

With *instance-based*, we identify those approaches where every text from the training corpus is considered individually. Each is treated as bearing a piece of information about its author. Each text is an *instance* of its style, and unknown texts are classified based on their relationship with this constellation of style samples.

In *profile-based* approaches, instead, all the available data from every author is grouped in a single file. This file is then used to extract a comprehensive representation of the author's style. This representation is the author's *profile*, and every unknown text is compared with it.

We will briefly outline the main strengths and weaknesses of the two approaches.

2.4.1 Instance-Based Approaches

In *instance-based* approaches, every text is considered individually, containing a sample of the author's style. This approach allows observing some aspects not accessible to *profile-based* approaches.

First, *instance-based* approaches can capture internal differences in the authors' style. For example, the anonymous text may be similar to part of the production of its author but quite far from the average sample. These inhomogeneities are expected when the corpus contains documents of different genres. In a corpus of

emails containing both personal and work messages, personal mail of authors with mostly work samples in the training corpus (or *vice versa*) may be far from the mean. Having all the samples from each author separated may help in such circumstances.

A second benefit is the opportunity to better take into account structural features. Since every text is independent, it is possible to better account for its macroscopic constituents. These features can be helpful in short texts where other features may be underrepresented. In the previous example, the consideration of greetings and farewell clauses in the messages [162] may help identify the author's style across different contexts.

A third important feature of *instance-based* approaches is the ability to leverage the capabilities of modern classifiers. The many samples per author are immersed in a high dimensional space where tools like ANN [55, 117] or SVM [40, 109] can work efficiently. These classifiers are also good at working with sparse and noisy data. This ability, in turn, means that it is possible to avoid or reduce the need for feature selection that is always arbitrary.

The use of classifiers like ANN or SVM makes fast the attribution of each text in the test set. The time spent to assign all the texts in the test set is mainly proportional to their number. On the other hand, the training phase of these classifiers may be lengthy for the many parameters that need to be optimised.

Another limitation of *instance-based* approaches is due to imbalances in the training corpus. Imbalances in the number and total length of the training samples may affect the classification output. If some texts are much longer than the others, they must be split to avoid biases. Splitting the training samples is also the primary option for authors with few texts. For the classifier to work, every class must be represented by several samples not too small. Split texts may however pose another kind of problem, e.g. with structural features.

When some authors have only a few short samples (even only one), finding a representation using *instance-based* approaches might be difficult. In these cases, an option is to use the training data more than once through text resampling [141]. These approaches indeed suffer when learning on short samples. Many studies are devoted to analysing the effect of short texts or chunks [61, 128]. The approaches' performance decreases with the training chunks' size, especially for sizes about the hundreds of words or below.

2.4.2 Profile-Based Approaches

Profile-based approaches offer partially complementary features. These methods were the first to be developed as they do not require the powerful classifier that became available only in recent years.

Each author's training material contributes to getting a single profile. Often this is obtained by simply joining all the training texts in a single large file per author. Then stylometric features are extracted, disregarding the original differences between the training texts. Even if this approach accounts well only for an author's style that is constant over the samples, a relevant benefit is the simplicity of the training phase.

The training phase for this group of approaches often consists of the sole profile extraction, often reduced to the feature extraction from the author file. Using a single profile reduces the noise over the extracted features as all the information is used at once. For attribution, very simple classifiers are used, often some *first nearest neighbour*, requiring little or no training.

The use of simpler classifiers often means the inability to treat many sparse features. Thus some feature selection is needed to reduce the feature set to a suitable size. Also, simple classifiers like *first nearest neighbour*, having no training phase, require comparing the anonymous text with all authors' profiles during attribution. This comparison may be pretty expensive, and the computational cost of the attribution grows like the number of texts in the test set times the number of authors.

Another feature of *profile-based* approaches is their relative resistance to imbalances in the training corpus. First, the text lengths in the training corpus have no effect. Long and short texts of an author are joined together without splitting nor the presence of short, noisy chunks. However, imbalances in the author length distribution may introduce strong biases. This is the case, for example, of the *Common N-gram* approach [75]. In this approach, only the L most frequent N-grams are considered.If at least one author has a short profile containing less than L different N-grams, the attribution strongly favours it [140]. This bias correction is not trivial as it is easy to go for the other excess [53] and favour authors with longer profiles. Due to the widespread use of balanced corpora in authorship attribution tasks, problems linked to class imbalance went substantially unnoticed for many decades.

This strength against imbalances is amplified when working on corpora with many authors (thousands to hundreds of thousands). Koppel *et al.* [81] observed in 2011 how similarity-based approaches work better than classifiers when the number of classes is large. Narayanan [107] pointed out one of the reasons as the difficulty of configuring complex classifier models with more parameters given a small number of training examples in each class.

Chapter 3

The Continuous Poisson-Dirichlet – Discrete Probability Approach

The approach we present in this thesis tackles the task of authorship attribution using a Maximum Likelihood principle. We derive the likelihood of an author given a text from a PD process introduced in chapter 1.

We represent each author $\mathcal{A}$ as a Poisson-Dirichlet (PD) process emitting tokens. Given an author's PD process, we can determine the conditional probability of any sequence to be its subsequent output. Then, the conditional probability that some anonymous text $\mathcal{B}_j$ is the output of the process of a known author is the likelihood $\mathcal{L}(\mathcal{B}_j \mid \mathcal{A}_i)$.

The categories introduced in chapter 2 can help frame our approach in the stylometry tradition. We will use a probability profile-based approach relying only on lexical tokens. The result is a language-independent approach, both in capabilities and outcomes, that offers state of the art results on small balanced datasets and large imbalanced corpora.

For every author, we create a profile containing the tokens they used and the parameters of their PD process. We use the complete corpus of their known production at once to determine the probability of an anonymous text.

One of the advantages of our method is its simplicity compared to many alternatives. It relies on lexical token unigrams without PoS tagging or classifiers. Our model is closer to topic modelling; however, it requires no statistical sampling due to the absence of latent variables. Thus, the attribution is fast and deterministic.

At first sight, our approach may seem an extreme simplification of approaches like the DADT-P, see section 2.3.4. We remove document topics and place every author

in one-to-one correspondence with a topic. Instead, we will show how, thanks to the superior expressiveness of the PD process, we obtain better results than DADT-P and related approaches on the most challenging corpora, see chapter 7.

The key to the simplicity of our approach and its success lies in a fundamental assumption. We will use the form of the PD process with a continuous base probability distribution from Eq. (1.33) (Continuous Poisson-Dirichlet, CP). However, we will use a discrete base probability for the tokens (Discrete Probability, DP). For this reason, we call our approach 'CP-DP'.

We will discuss the theoretical implications of this assumption in chapter 5. Furthermore, its effect will be evident in chapter 7 from the comparison with other approaches. Finally, we will comment on the general feasibility of avoiding this assumption while discussing some technical aspects of our approach in chapter 8.

We breakdown our approach into three main phases:

1. feature extraction, where we normalise and tokenise all input documents;
2. likelihood estimate, where we compare the unknown texts with the profile of the authors;
3. attribution, where we propose the most likely author.

In the feature extraction phase, we measure the documents. Starting with a stream of characters, we project the text in the space of tokens. However, this projection is not unique. Even considering only lexical features, there are many available options.

In this work, we will consider three groups of features: word tokens, N-grams, and LZ77 sequences. We will present these three projections in chapter 4 as the first hyperparameter of our model. We will note how different corpora require different variables for a fruitful representation.

In section 3.1, we will discuss the estimate of the likelihood. This procedure is straightforward but requires estimating discount and concentration parameters for every author's profile and evaluating the base probability. The choice of the base probability normalisation introduces our second and third hyperparameters discussed in chapter 5.

Here, we introduce an essential step for completing our task. We will split the documents into fragments and compute a separate conditional probability for each one. This step will make our approach more resilient against corpus imbalances. It is possible to apply the slicing only to the unknown texts or also to the authors' corpora. The fragments' size is our fourth hyperparameter. We will discuss the effects of slicing in chapter 6.

The last phase is the proper attribution presented in section 3.4. It is not trivial when we must merge the information from different fragments. In this case, we will

discuss two different views on attribution. In the first, we assume the fragments are independent, multiply their probabilities, and choose the author via Maximum Likelihood (ML attribution). In the second, we attribute each fragment via Maximum Likelihood. We assume every text is single-authored. Hence, we adopt a Majority Rule choosing the most likely author of the most fragments (MR attribution).

We will check the performance of our approach on four different corpora described in section 3.2. First, we will use them to evaluate the different choices in the attribution technique. Then, we will use the same corpora to compare our approach with others in chapter 7.

Figure 3.1 presents a schematic view of our approach with its different phases.

3.1 Estimating the Likelihood

We estimate the likelihood that some reference author wrote a text through a PD process that, at each step, outputs a token. The likelihood is the conditional probability that the tokens in the anonymous text come from the same process that already produced the corpus of the reference author.

We are interested in the conditional probability of a text $\mathcal{B}$ given the whole production of an author $\mathcal{A}$. We represent $\mathcal{B}$ as a sequence f of tokens and $\mathcal{A}$ by their profile. The profile contains the sequence A, the union of all their texts, and parameters α_A and θ_A. We can write the product of the probability in (1.15) for every token in f as:

$$P(f \mid A, \alpha_A, \theta_A, P_0) = \frac{\prod_{j=k'}^{k'+k-1}(\theta_A + j\alpha_A)}{\prod_{j=n'}^{n'+n}(\theta_A + j)} \prod_{j=1}^{k'+k} Q_j = \\ = \frac{(\theta_A + k'\alpha_A \mid \alpha_A)_k}{(\theta_A + n')_n} \prod_{j=1}^{k'+k} Q_j \quad (3.1)$$

Where we are using primed variables to refer to the reference author and the Pochhammer symbols introduced in section 1.2.2 Please note that when we consider unique tokens, they are unique to the whole sequence. The k unique tokens in f counts only those absent in A. In other terms, k is the cardinality of $\{y_j \in f\} \setminus \{y_j \in A\}$.

The parameters α_A and θ_A are specific to the author's process and have to be estimated. We present the way to estimate the parameters in section 3.3. The term Q_j depends on whether token y_j is present in A or not:

$$Q_j = \begin{cases} (1-\alpha)_{n_j-1} P_0(y_j)\delta & \text{if } y_j \notin A \Leftrightarrow j > k' \\ (n'_j - \alpha)_{n_j} & \text{otherwise.} \end{cases} \quad (3.2)$$

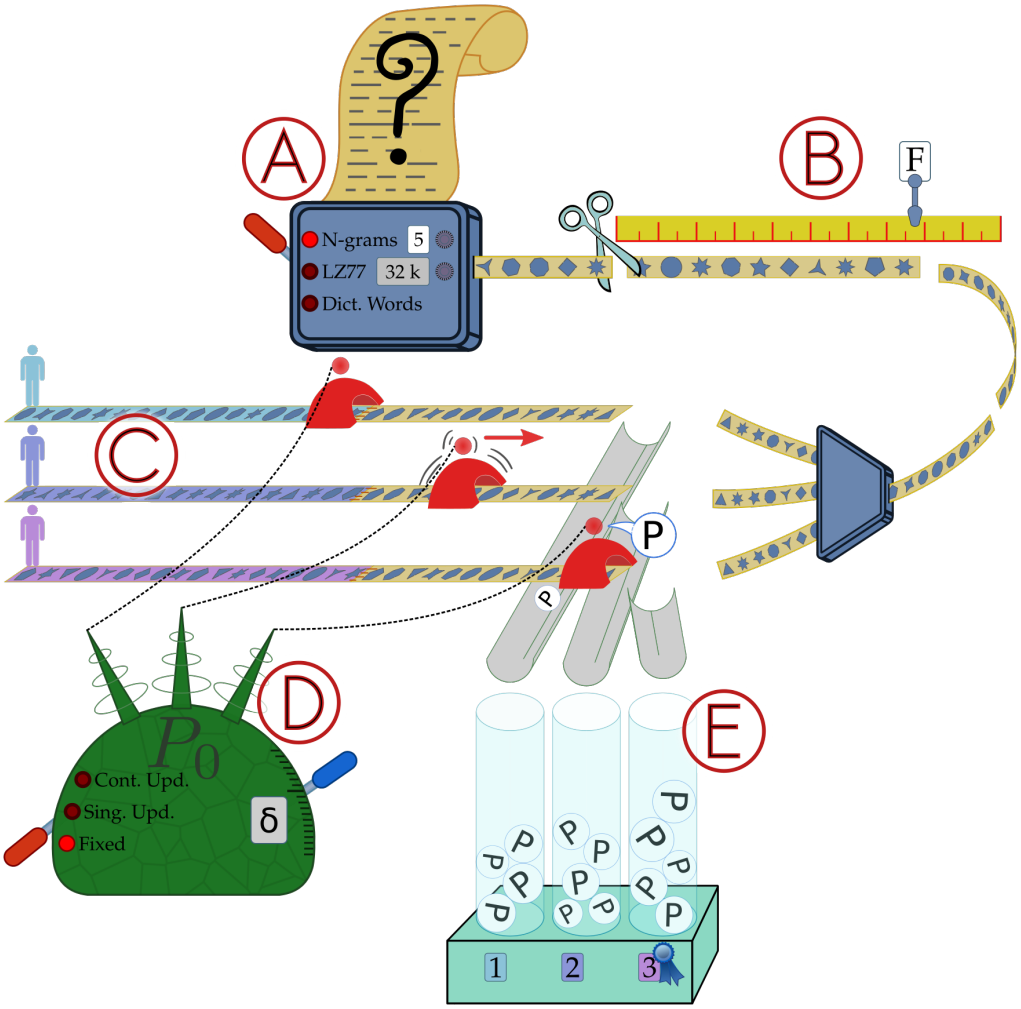


Figure 3.1. Schematic view of the CP–DP approach. A — the unknown text is preprocessed following one of the techniques described in chapter 4. B — the stream of tokens is divided into fragments of length F according to the considerations from chapter 6. C — a copy of the fragment is attached to the end of each author's production, and its conditional probability is computed; see sections 3.1 and 3.3. D — The PD processes of the authors are equipped with a suitable P_0 presented in chapter 5. E — the conditional probabilities of all the fragments are combined to propose the most likely author as described in section 3.4.

In the first case of Eq. (3.2), we introduced the hyperparameter δ. We pose $\delta > 0$, and equal for all authors and documents. The reason for its introduction will be discussed in section 5.2.

For computational purposes, we can rewrite Equations (3.1) and (3.2) using Gamma functions. This is the form that, in logarithmic space, we used in the code:

$$P(f|\mathcal{A}, P_0) = \frac{\alpha^k \Gamma\left(\frac{\theta_A}{\alpha_A} + k + k'\right) \Gamma\left(\theta_A + N\right)}{\Gamma\left(\frac{\theta_A}{\alpha_A} + k'\right) \Gamma\left(\theta_A + N' + N\right)} \prod_{j=1}^{k'+k} Q_j \qquad (3.1\ bis)$$

$$Q_j = \begin{cases} P_0(y_j) \frac{\Gamma(n_j - \alpha)}{\Gamma(1-\alpha)} & \text{if } y_j \notin A \Leftrightarrow j > k' \\ \frac{\Gamma\left(n'_j + n_j - \alpha\right)}{\Gamma\left(n'_j - \alpha\right)} & \text{otherwise.} \end{cases} \qquad (3.2\ bis)$$

In eq.(3.1 *bis*) we grouped all the conditions over A, α_A and θ_A as a condition over $\mathcal{A}$ to lighten the notation. This probability $P(f \mid \mathcal{A})$ of the sequence f to be the next output of the process that produced the sequence A, is thus the likelihood, what we called $\mathcal{L}(\mathcal{B} \mid \mathcal{A})$.

3.2 Corpora

In this study, we considered different corpora to answer different questions. We chose literary corpora to investigate the effect of different languages while using long texts from a few tens of authors. On the other hand, we used two English corpora of informal texts: one of email, the other of blog posts. We use the informal corpora to evaluate the ability of our method in more challenging cases with thousands or hundreds of thousands of short texts.

The Literary Corpora. Our first question was: does our method work on different languages? For example, changing language, a word may have a single form or many, depending on its semantic role. Adjectives have one form in English (*red*), four in Italian (*rosso*, *rossa*, *rossi*, *rosse*), ten in Polish (*czerwony*, *czerwona*, *czerwone*, *czerwonego*, *czerwonej*, *czerwonemu*, *czerwoną*, *czerwonym*, *czerwonych*, *czerwonymi*). This may introduce differences in favour of one or another language as investigated in the works of Eder and collaborators [42, 125, 44, 43]. Our model is a good candidate as we designed it based on token frequencies and relies on no semantic tagging.

To investigate this, we considered three corpora of literary texts: 1. English, 2. Italian, 3. Polish. From every corpus, we excluded authors that had only one book.

The English corpus consists of 439 books from 44 authors active in the UK around the end of the 19$^{\text{th}}$ century. It is derived from the one used in [88] selecting

only native British English speakers from the UK alive in 1894 to ensure a relative linguistic uniformity. We removed some books more to avoid a strong unbalance between authors with only a couple of texts and authors with dozens. When using relatively few texts, this unbalance may mask biases in the attribution if prolific authors are preferred no matter what is in the text. We decided to take all the books for authors with less than ten books while, for those with ten books or more, we kept the nine books closest to the author's average length plus one every ten at random to maintain the overall ranking in the number of texts.

The Italian corpus contains 171 books from 39 contemporary Italian authors. It is derived from the one used during the workshop "Drawing Elena Ferrante's Profile" held in Padova, Italy in September 2017 extended as in [89] but without the writings from Elena Ferrante. The author behind this *nom de plume* was never officially identified, and the inclusion of her books would have added a significant source of noise.

The last of the literary corpora contains 100 books from 34 Polish authors. It is derived from a benchmark corpus of 100 Polish novels, covering the 19th and the beginning of the 20th century. We removed one book that is the only example from Magdalena Samozwaniec. Thus, we used 99 novels by 33 authors (1/3 female writers, 2/3 male writers). The Computational Stylistics Group at the Institute of Polish Language[1] prepared the corpus for stylometric analysis.

The Informal Corpora. The second question was: is our approach practical also on unbalanced datasets, possibly large and full of short texts? To test under these circumstances, we included two corpora of informal writing: the Email and Blog corpora. The choice fell on these two corpora with partly complementary features and already used as tests for other approaches.

The Email corpus is one of the corpora used during the PAN'11 contest [11]. The corpus was developed based on the Enron Email corpus[2] and divided into sections to account for different common attribution and verification scenarios. We selected only one pair of the twelve separate training and test collections – of which six are dedicated to author verification. We use the "Large" training sets provided for authorship attribution containing 9337 documents by 72 different authors. As a test set, we chose the one containing texts written only by the authors in the training set while leaving the other one (also containing texts written by around 20 other "new" authors) for future analysis.

The tasks at PAN'11 intended to reflect a natural task environment, so the curators included texts in languages different from English or automatically gen-

[1] https://github.com/computationalstylistics/100_polish_novels
[2] http://www.cs.cmu.edu/enron/

erated. The curators automatically replaced personal names and email addresses with <NAME/> and <EMAIL/> tags, respectively. We decided to put ourselves in the same conditions as the PAN competition and did not look for the elements that eluded this cleaning operation. We replaced the XML tags with "ANAME" and "EMAIL" strings. Under any other respect, the texts are guaranteed to be identical to the original. The original curators of the corpus had determined authorship based on the "`From`" email headers. The curators tried to match all the addresses belonging to the same individual; however, they acknowledged some errors.

The last and largest dataset we considered is the Blog corpus. This is a collection of 678,161 blog posts by 19,320 authors took from [131][3]. This corpus is valued for its size and, even if it has been around for quite a long time, it is still in use to test novel stylometric approaches [126]. Before focusing on the whole corpus, we devoted our analysis to a subset of the 1000 most prolific authors. This is the subset used in [134, 133] containing the top 1,000 authors by the number of blog posts.

It is, however, not easy to find references in performance on this corpus. The corpus is quite large and contains texts of very different lengths. For many *authorship attribution* approaches this requires the corpus to be reduced [65, 111, 126]. This necessity is due to limitations in handling many authors/texts, to the need for texts of a certain minimum length or many examples from every author.

We found some oddities in the corpus, like having the same blog appearing twice under different authors ids. However, we considered this part of the challenges of working with a natural environment and decided not to fix such issues. Likewise, we kept all the posts even if around 0.5% have at most ten characters, and about 11.5% have no more than 100.

Every text is encoded using a single-byte encoding ISO 8859-1 for Italian and English texts, ISO 8859-2 for the Polish ones. While this forces transliterating some rare off-encoding characters (e.g. a Greek quotation in an English book), it also ensures the correct behaviour during feature extraction.

The length of the texts and the production of the authors vary widely across the different corpora. In figure 3.2 we show, in the top panel, the size-rank plot for texts. The difference between literary and informal corpora is evident. Informal texts are not only shorter (blog posts) or much shorter (email) than books, their size varies a lot more. For literary corpora, the standard deviation of book sizes is about 60% of the average book size. In the case of informal corpora, the fluctuations are about twice the mean for blog posts and thirteen times the mean in the case of email.

[3] Available for download from `u.cs.biu.ac.il/~koppel`

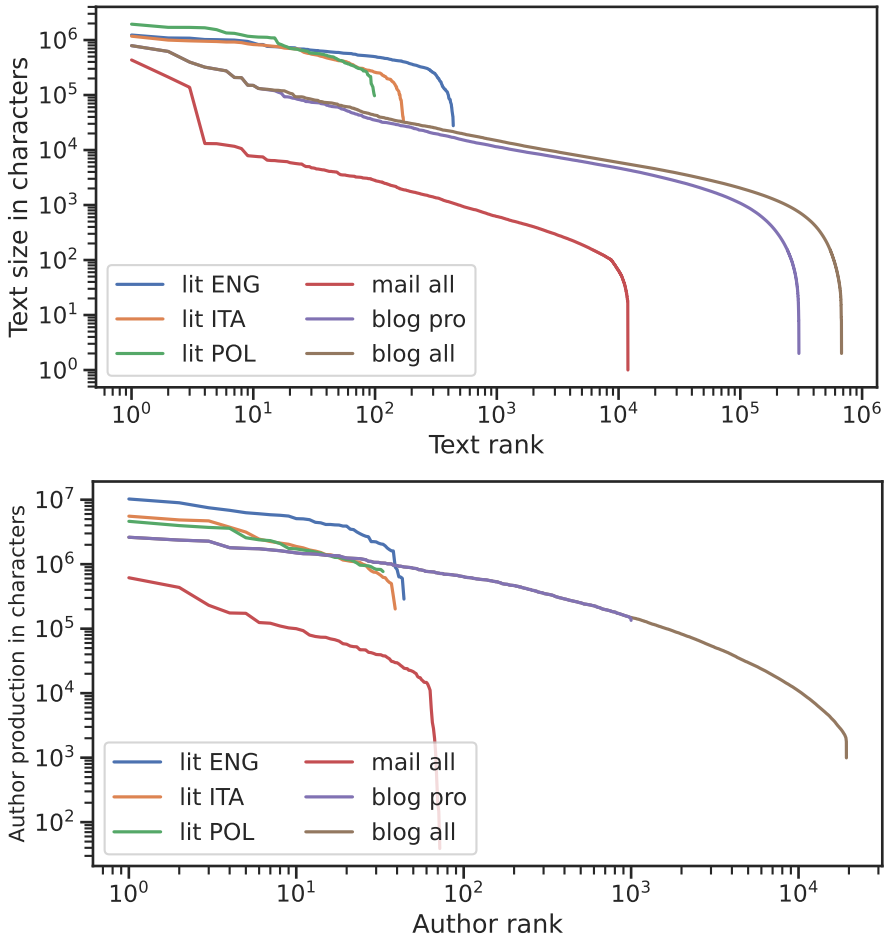


Figure 3.2. Size of the texts and production of the authors in bytes. We measured the length after reducing all texts at a single-byte encoding.

Looking at the authors' production in the bottom panel of figure 3.2 we still notice large fluctuations in size. The deviations range from 0.6 to 2.5 times the mean in the case of the Polish and the complete Blog corpus respectively. Moreover, for every corpus, we notice that there is at least some text longer than the whole production of some author. These wide variations suggest that using fragments of texts and slices of authors might reduce biases and bring all texts and authors to a common ground.

3.3 Estimating the Discount and Concentration Parameters

To compute the probability from Eq. 3.1, we need the discount and concentration parameters. We estimated them based on the author's sequence only. The simple form of the probability of a sequence in Equation (1.33) suggests an easy way to evaluate the parameters α_A and θ_A given the sequence A of an author. Given the $n_1, \ldots, n_k$ multiplicities of the k different tokens in A, we can find α_A and θ_A following a maximum likelihood principle. In practice, as done for example, in [90], we chose α_A and θ_A that maximise the probability of the sequence.

The discount and concentration parameters estimate is independent of the base probability distribution. Indeed, looking at Eq. (1.14), α and θ determine the p_i, the P_0 determines the y_i that appear as arguments of the δ. The $P_0(y_i)$ are fixed and contribute with a factor independent from α_A and θ_A. Thus, our task reduces to the search for the pair of parameters that maximises the probability of the partition of n elements in k classes:

$$(\alpha_A, \theta_A) = \arg\max_{(\alpha,\theta)} \frac{(\theta \mid \alpha)_k}{(\theta)_n} \prod_{j=1}^{k} (1-\alpha)_{n_j-1} \tag{3.3}$$

We search for the maximum using the steepest gradient ascent. To find the values of α and θ that ensure the maximum probability, we iterate the following system until convergence:

$$\begin{cases} \alpha(t+1) = \alpha(t) + I_\alpha a_\alpha(t) \\ \theta(t+1) = \theta(t) + I_\theta + a_\theta(t) \end{cases} \tag{3.4}$$

For the gradient we need the derivatives of Eq. (3.3) with respect to α and θ. The gradient assumes a nice formulation in terms of the Gamma function logarithm derivative: the Digamma function ψ^0. We are searching for the maximum, i.e. when both derivatives are null. Thus, we do not need the exact form of the derivative, and a quantity proportional to it is enough.

For the derivative in θ we have:

$$\frac{\partial P}{\partial \theta} \propto \frac{\partial}{\partial \theta}\left[\alpha^k \frac{\Gamma\left(\frac{\theta}{\alpha}+k\right)}{\Gamma\left(\frac{\theta}{\alpha}\right)}\frac{\Gamma(\theta)}{\Gamma(\theta+n)}\right] =$$

$$= a^k \times \left[\frac{1}{\alpha}\frac{\Gamma'\left(\frac{\theta}{\alpha}+k\right)}{\Gamma\left(\frac{\theta}{\alpha}+k\right)} - \frac{1}{\alpha}\frac{\Gamma'\left(\frac{\theta}{\alpha}\right)}{\Gamma\left(\frac{\theta}{\alpha}\right)} + \frac{\Gamma'(\theta)}{\Gamma(\theta)} - \frac{\Gamma'(\theta+n)}{\Gamma(\theta+n)}\right] \times \frac{\Gamma\left(\frac{\theta}{\alpha}+k\right)}{\Gamma\left(\frac{\theta}{\alpha}\right)}\frac{\Gamma(\theta)}{\Gamma(\theta+n)} \propto$$

$$\propto \frac{1}{\alpha}\left[\psi^0\left(\frac{\theta}{\alpha}+k\right) - \psi^0\left(\frac{\theta}{\alpha}\right)\right] + \psi^0(\theta) - \psi^0(\theta+n) = a_\theta(\alpha, \theta \mid n, k) \quad (3.5)$$

where the last line is the only thing we need to find the maximum. Similarly for the α derivative we find:

$$\frac{\partial P}{\partial \alpha} \propto \frac{\theta}{\alpha^2}\left[\psi^0\left(\frac{\theta}{\alpha}\right) - \psi^0\left(\frac{\theta}{\alpha}+k\right)\right] + \frac{k}{\alpha} +$$

$$+ \sum_i r_i \left[\psi^0(i-\alpha) - \psi^0(1-\alpha)\right] = a_\alpha(\alpha, \theta \mid k, \mathbf{r}) \quad (3.6)$$

where r_i is the number of tokens with multiplicity i, that is, the number of partitions of size i, with $\sum_i r_i = k$ and $\sum_i i r_i = n$. These formulas allow a fast and accurate derivative computation for a precise search for the maximum.

In practice, we find that the plain gradient ascent is slow due to regions of little variation of the probability. Therefore, we chose to use the gradient ascent with momentum to improve the performance.

We treat α and θ as the coordinates of a unit mass point moving in a damping medium. Now a_α and a_θ become the components of the acceleration. A damping force proportional to the speed opposes the movement.

The new set of equations becomes:

$$\begin{cases} \alpha(t+1) = \alpha(t) + I_\alpha v_\alpha(t+1) \\ v_\alpha(t+1) = v_\alpha(t) - \eta v_\alpha(t) + a_\alpha(t) \end{cases} \quad \begin{cases} \theta(t+1) = \theta(t) + I_\theta v_\theta(t+1) \\ v_\theta(t+1) = v_\theta(t) - \eta v_\theta(t) + a_\theta(t) \end{cases} \quad (3.7)$$

where the I are scaling factors, a is an acceleration, and η is a dissipation term. The dissipation avoids the build-up of excessive momentum and undesired oscillations around the maximum.

We added momentum to improve the algorithm's efficiency over regions with small gradient values. We also added a second feature to speed up convergence in cases when the derivative changes sign from one step to the other, i.e. we passed the maximum. We switch to a bisection method that quickly narrows the search area. This allows for larger scaling factors and avoids passing by the maximum at full speed.

When the optimisation process leads out of the acceptable range of values ($\alpha \in [0, 1)$ and $\theta > -\alpha$), it is continued from a value close to the border (0.01 and

0.99 for α and $-\alpha + 0.1$ for θ). We allow this to happen a limited number of times. Most of the time, we reach unacceptable values due to the high absolute value of the derivatives and high increments next to the domain's borders. We reduce the momentum by increasing η and reducing the scaling factor to mitigate this effect.

In figure 3.3, we show a contour plot of the tokens' partition probability for a specific author changing α and θ. For example, we show here the case of George Alfred Henty (1832-1902), the author in the literary English corpus with the most significant amount of tokens using *Dictionary words*: $n = 4.44 \times 10^5$ tokens of which $k = 3.20 \times 10^4$ different. We find the maximum probability for $\alpha = 0.3343$ and $\theta = 740.3$ where $\log_{10} P^{\max} = -5459939$.

The optimisation using the steepest gradient ascent with momentum requires setting some parameters. We chose the starting values for α and θ to be respectively 0.3 and k, the number of different tokens. We take the base scaling factors over α and θ to be 10^{-4} and 1 roughly reflecting their relative magnitude. We halve the scaling factor every time the value proposed for one of the parameters exits the acceptable range. Finally, we set the initial damping to 0.1 and increase it only when failing on θ. We chose this rule because, when θ fails in the proximity of $\theta = -\alpha$, the probability is very steep.

The maximisation may not always converge. In that case, we use the closest acceptable value. This, in practice, happens only with α where we used the values 0.001 and 0.999. Even if a value of 0 would be acceptable, we force it to be slightly greater to avoid the degeneration to a Dirichlet Process.

In chapter 4, we present an overview of the results of the parameters optimisation changing variables. We present separate results for the different kinds of features. The three approaches produce different sets of tokens with different statistical properties and well-defined trends in the concentration and discount parameters.

3.4 Attribution

Once we estimate the likelihood for every text (or fragment) compared with every author (or slice), we must assign the text to a single author.

In case neither the text nor the author are divided into parts, the task is trivial. We assign each text f to an author $\mathcal{A}$ following a maximum likelihood principle, i.e. to the author such that $\mathcal{A} = \arg\max_A P(f|A)$.

A different point of view is to consider every author's process as a language model. The perplexity of a model on the test data is defined as an Equation.

$$PP(f) = a^{-\frac{1}{n}\sum_{i=1}^{n} \log_a P(x_i)} = a^{-\frac{1}{n}\sum_{i=1}^{n} \log_a P(x_i|\mathcal{A},\{x\}_1^{i-1})} = a^{-\frac{1}{n}\log_a P(f|\mathcal{A})} \quad (3.8)$$

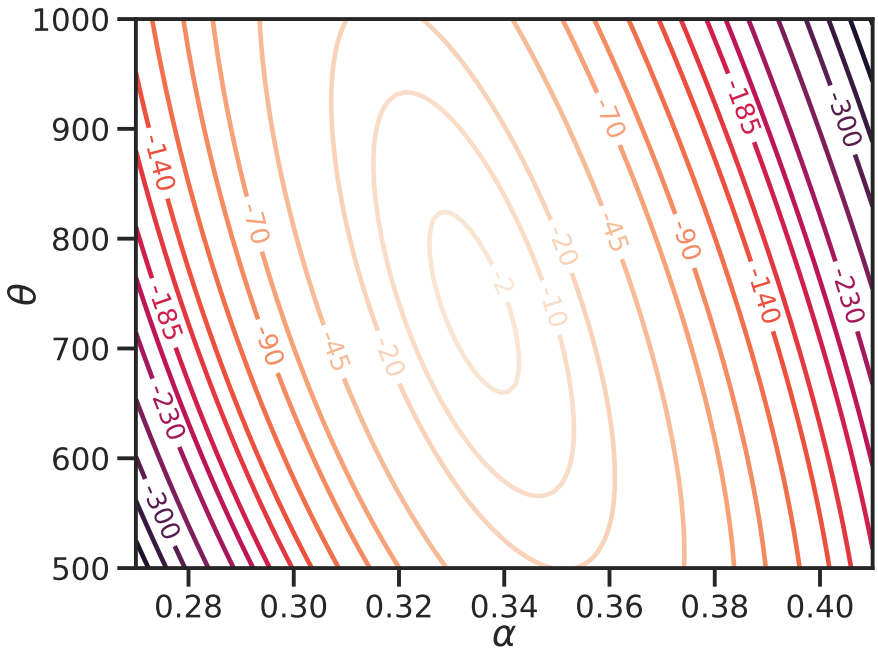


Figure 3.3. Typical shape for the probability of the partition varying α and θ. This contour plot shows the logarithm values of the probability of the partition in k classes of the n tokens of George Alfred Henty. The values in the plot are relative to the maximum where $\log_{10} P^{\max} = -5459939$. Points on the innermost contour line denote choices of parameters leading to probabilities one-hundredth of the maximum.

The process that gives the maximum probability is the one that better models the sequence. Thus, we are selecting the model with the lowest perplexity.

When we split the text into fragments of size F, we can still resort to a maximum likelihood principle or follow a majority rule.

Maximum Likelihood If we choose a maximum likelihood approach, we assume independent fragments. In this case, we approximate the likelihood as:

$$\mathcal{L}(\mathcal{A}|f) \approx \prod_{i=1}^{\frac{n}{F}} P(f_i|\mathcal{A}) \tag{3.9}$$

From the point of view of the author-as-a-language-model, using fragments, we do not assume independence. We are just forcing our model to reset every F steps. As discussed when considering short fragments and authors, this cuts the adaptation of the process of the reference author to the unknown text due to the previous fragments.

In case the reference author is also sliced in s parts, we need to find a way to estimate the likelihood for the author as a whole. After this, we can proceed as in (3.9). For the fragments' likelihood estimation, we considered three different options. We can consider only the slice that offers the highest likelihood for every fragment. Possibly prone to noise, this option represents the idea that an author may exhibit different styles, and only a part of their production will match the anonymous text. We can instead take, for every fragment, the geometric mean of the likelihoods for every slice. Given the relative magnitude of the probabilities, taking the arithmetic mean would be the same as taking the maximum. Doing all the computation in logspace is natural to consider the geometric mean. This solution is not proportional to the case of the full author. The process is not linear, and the token distribution within the author corpus is uneven.

We can consider an intermediate weighted profile approach. This approach exploits further information than the maximum value of the likelihood between the fragments f_i and author slices A_j. In particular, we wish to use the information coming from all the slices of a given author in the corpus, trying at the same time to reduce noise. We rank all the s_j slices of a given author in descending order according to the probability $P(f_i|A_j)$ of f_i. We then construct a weighted average for each author that we use as a measure of the likelihood:

$$P(f_i|A) = \frac{\sum_{j=1}^{s} \frac{1}{j} P(f_i|A_j)}{\sum_{j=1}^{s} \frac{1}{j}} \tag{3.10}$$

Majority Rule In this frame, we consider each fragment as carrying part of the information about the authorship of the entire text. Let the text f be composed of

n fragments. We first attribute each fragment to an author, then (assuming we do not have any *a priori* information about the higher reliability of one fragment or one other), we attribute the whole text f to the author to which the majority of the fragments f_i point.

As done before with whole texts, we attribute each fragment via maximum likelihood. The probability of each fragment is given if author corpora are not sliced or estimated in the ways proposed above.

It should be noted that in a majority rule approach, a situation of parity can occur, especially in the case of texts divided into few fragments. We attribute the same maximum number of fragments to different authors. In a case of parity, we consider the attribution failed. Even if one of the top authors is correct, we failed to point out the right author. In practice, this happens almost only when the document is divided into a few tens of fragments. For the literary English corpus, using 9-grams and with 1000 characters fragments, we have a 1.1% of ties. We excluded counting half successes to avoid artificial favourable conditions.

Our approach does not need training. Instead of tuning or sampling parameters, we only need to optimise and set four hyperparameters. The four hyperparameters are:

- the choice of variables for feature extraction;
- two for the normalisation of P_0 (update rule for the denominator and δ coefficient);
- the use and size of the fragments.

The following three chapters will address the effects of the different choices and offer some guidelines.

To keep the exposition fluent, we report only a part of the graphs illustrating the results in the main text instead of presenting all corpora for every section. We selected the most relevant or insightful ones gathering the others in Appendix C. Moreover, to keep the graphs clean, we focus on the section's topic. When commenting on the effects of some hyperparameter, e.g. a specific form for the base probability distribution, we report only the best results under different choices of the other hyperparameters, e.g. the kind of variables.

When evaluating the effect of the hyperparameters, we will consider a single figure of merit to evaluate or results. We will focus on the overall fraction of documents assigned to their author, calling this fraction 'score'. We evaluate the score using a leave-one-out scheme on the entire corpus. Every document in the corpus, in turn, is considered the only unknown. We will use more detailed measures when comparing our approach with others.

Chapter 4

Choosing the Variables

In the preprocessing phase, we manipulate texts to produce sequences of tokens suitable to be treated as the outputs of a PD process. We are looking for the best variables to project our texts. There could be many different feature extraction approaches, but we focus mainly on three: 1. *Dictionary Words*, 2. *Overlapping Space-Free N-grams* and 3. *LZ77 Sequences*. Each one of these methods has an associated normalisation.

Two steps are common to the three methods. First, we replace all the newlines with spaces. Second, we replace all the Unicode punctuation with its ASCII equivalent. This normalisation derives from not relying on white space or possibly spurious features. The shape of punctuation may depend more on the editor's taste (in the case of literary texts) or the software used (in informal texts). For example, using "plain" or “smart” quotes often depends on the software or OS used. Identifying an author because of their consistency in using the same software is of no interest in this context.

Dictionary Words. This approach gives as result tokens similar to everyday words found in *dictionaries*. We replace all non-alphabetical characters with spaces and split the sequence in tokens using spaces. This allows the presence of tokens like “doesen” that, even if it is not a regular English word, is the expected behaviour also of other tokenisation techniques, e.g. [161].

Overlapping Space-Free *N*-grams. In the second case we consider *space-free character N-grams* as defined in [81]. These are strings of length N that may include spaces only as first or last characters. We consider all the overlapping N-grams and keep all the punctuation as in [81]. This choice discards words shorter than $N-2$ and gives more weight to longer ones.

LZ77 Sequences. The last tokenisation technique uses a derivation of the *LZ77* algorithm [164] to extract repeated sub-sequences from the processed sequence. Every sub-sequence is a token. Tokens extracted this way lose their direct correspondence to natural words, as they can be parts of words or entire sentences. In this case, the length is free to vary according to the repetitions in the text, and we include spaces and punctuation. The idea is to capture stylistic markers as idiosyncratic expressions or a preferred way to organise thoughts.

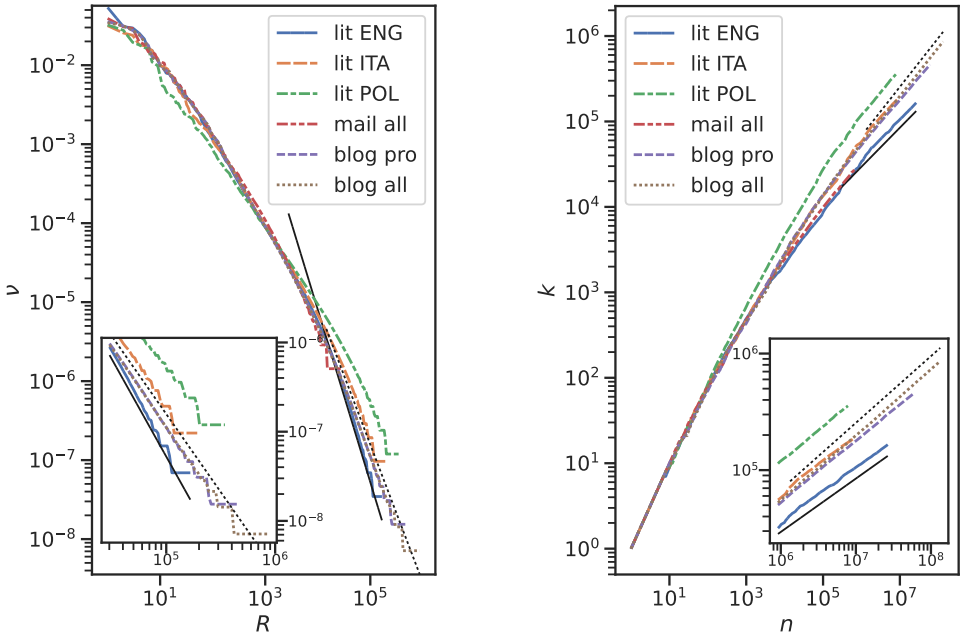
4.1 Dictionary Words

Using *Dictionary words*, we keep all the alphabetical characters and remove punctuation and numbers. In table 4.1 we report the fraction of characters in the original text that appears in the tokenised sequence. We hand-checked some of the texts with a suspicious low fraction of characters preserved. These texts mainly contained large tables with numbers, dashes used for tabulation, other typographical elements rendered as non-alphabetical characters in digitisation. We found even ASCII art reproducing greyscale images as text in blog posts.

Table 4.1. Characters in the tokenised sequence for every character in the original text and median length of the tokens. The micro average is computed on the whole corpus at once, the macro average is the average over the per text fraction of characters preserved. We report the median word length and the Median Absolute Deviation.

Corpus		Average		Word
		Micro	Macro	length
Literary	Polish	0.946	0.947	5±2
	Italian	0.968	0.968	4±2
	English	0.963	0.961	4±1
Email		0.847	0.917	4±2
Blog	all authors	0.944	0.929	4±1
	prolific	0.944	0.931	4±1

The frequency rank of the tokens in the corpus shows the expected power-law tail – i.e. Zipf's law, see Fig. 4.1a – usually observed in texts (e.g. see [150]). The same is true also for the number of different tokens after reading n tokens, i.e. Heaps' law, see Fig. 4.1b. In this case, we refer to a sequence obtained by random concatenation of all the texts in a corpus.



(a) Zipf's law **(b)** Heaps' law

Figure 4.1. Zipf's law and Heaps' law for the different corpora using *Dictionary words*. Straight lines in the Heaps' law plots show functions of the form $f(x) = ax^{\beta}$, with the highest and lowest fitted exponents β equal respectively to $\beta = 0.572$ (Blog – all authors) and $\beta = 0.458$ (literary English). Other values are $\beta = 0.517$ (literary Italian), $\beta = 0.528$ (Blog – prolific), $\beta = 0.537$ (literary Polish), and $\beta = 0.545$ (Email). Straight lines in the Zipf's law plots show functions of the form $f(x) = ax^{-\alpha}$, where the exponent α is equal to β^{-1} for the highest and lowest βs considered above. Note that the frequency-rank plots deviate from a pure power-law behaviour and the correspondence between the β and α exponents is valid only asymptotically. The points closer to this limit are shown in the inset.

All corpora exhibit Taylor's law, the deviation σ_k from the mean number of different tokens per document k grows following the expected power-law. We report in figure 4.2 the curves for all corpora.

For later comparison with other features we report in table 4.2 the average fraction of *hapax* and *dis legomena*, tokens appearing only once or twice, over the number of different tokens. If a text has many words appearing only once, the number of words used by the reference author and entirely new ones strongly influence its probability. We notice the difference between literary and informal texts and how this increases considering the macro average. The many short texts in informal corpora are less likely to contain the same token twice. In this corpora, the use of author or text-specific tokens will have a more substantial effect.

Finally, we show in Fig. 4.3 the fluctuations of the tokens' frequency across different authors. This may help us understand the tokenisation techniques' different behaviours when using long fragments. We notice that as the fluctuations decrease with the frequency, higher ranks, their relative amplitude grows roughly as a power-law of the rank. This growth is a marker that rare words are unevenly distributed among authors and may carry helpful hints on authorship.

In figure 4.4 we show a KDE plot for the joint distribution of α and θ over the different corpora when using *Dictionary words*. A negative correlation between the two parameters is evident in literary corpora: a high value of α or θ conjures many different tokens. On the other hand, given the number of different tokens, if the distribution of tokens among classes requires a high α (see Eq. (3.6)), this implies a small θ and vice versa. Correlation coefficients span from -0.391 for the English corpus to -0.625 for the Polish one.

In the case of informal corpora, this is less evident due to the wider variety of authors' production. Authors with a smaller production allow for wider distribution. This correlation is evident when Considering only the prolific authors in the Blog corpus. Correlation coefficients span from -0.200 for the Email corpus to -0.412 for the prolific bloggers.

When considering the whole production of every author, using *Dictionary words*, the parameters optimisation converge for all the authors in all corpora. We report the best scores for every corpus in table C.1.

We present the attribution scores using *Dictionary words* in figure 4.5. We excluded the complete Blog corpus due to its size.

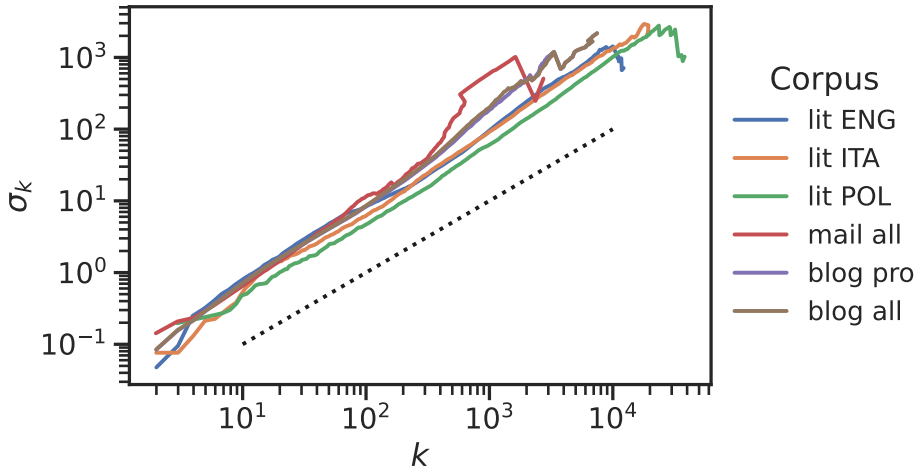


Figure 4.2. Taylor's law using *Dictionary words*. The dotted line is a power law with exponent one provided as a guide for the eye. The decreasing tract at the end of some curves is due to the reduced number of texts reaching very high values of k.

Table 4.2. Fraction of *hapax* and *dis legomena* using *Dictionary words*. The micro average is computed on the whole corpus at once, the macro average is the average over the per text fraction of *hapax* or *dis legomena*.

Corpus		*Hapax legomena*		*Dis legomena*	
		Micro	Macro	Micro	Macro
Literary	Polish	0.636	0.651	0.150	0.147
	Italian	0.584	0.593	0.154	0.152
	English	0.489	0.499	0.163	0.163
Email		0.806	0.867	0.119	0.093
Blog	all authors	0.736	0.799	0.136	0.118
	prolific	0.738	0.807	0.134	0.112

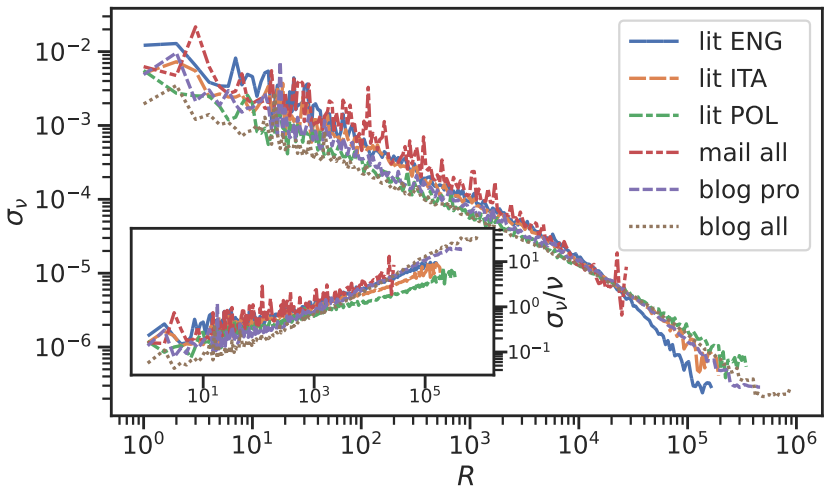


Figure 4.3. Fluctuations in token frequency across authors using *Dictionary words*. The main graph shows the standard deviation across authors of token frequencies, ordered by global frequency. The inset shows the growth with the rank of the relative amplitude of the deviation. Most common tokens have low relative fluctuations suggesting similar use across all authors, while the less common are more author-specific. Linear fits on the plot suggest exponents in the range $[0.3, 0.6]$.

4.2 Overlapping Space-Free *N*-grams

The second kind of variables we considered are the *Overlapping Space-Free N-grams*. The choice of *OSF N-grams* may require a bit of discussion. Compared to Dictionary words, *OSF N-grams* we discard all the words shorter than $N-2$ (or $N-3$ if followed by punctuation). Longer words, instead, may contribute with more than one N-gram. They will appear without their prefix or suffix, thus recovering something similar to the words' root. Choosing to retain only *Space-Free N-grams*, where a space can be only the first or last character, dramatically reduces the number of available N-grams.

We discard all the N-grams that bridge two or more words. Preliminary results showed that retaining all the N-grams negatively affects the results. Usual N-grams – for N not too small – are dominated by representations of word pairs that occur in a greater variety of forms. To capture this kind of information, we will use *LZ77 sequences*. *Space-Free N-grams* capture information at the word and sub-word levels.

This can be clarified with an example. Let's consider the two following texs:

1. `I love my penguin.`
2. `Look at the penguins!`

Using *OSF 8-grams* both texts are reduced to a few tokens:

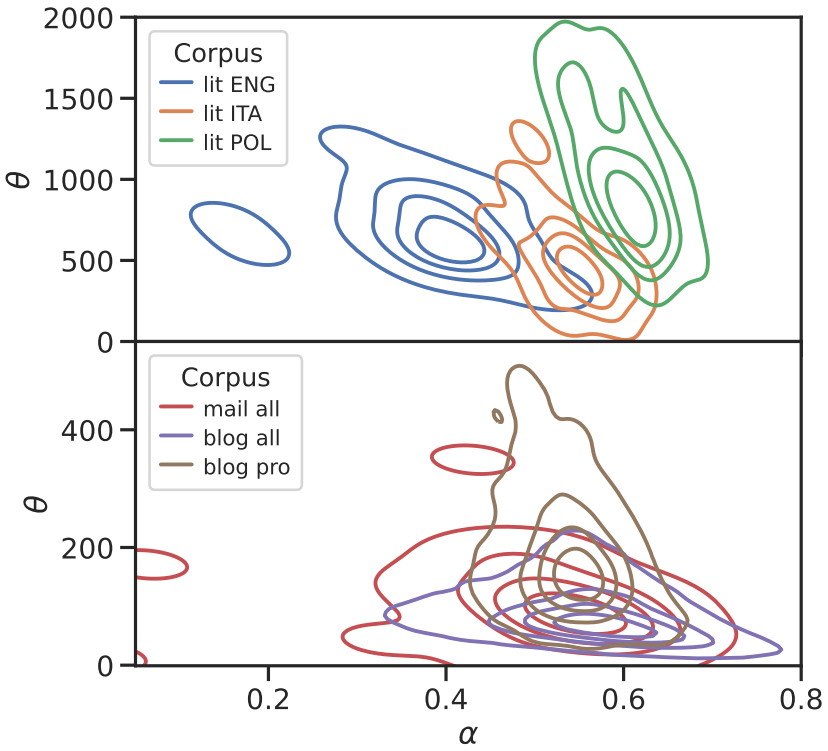


Figure 4.4. KDE plot for the join distribution of α and θ using *Dictionary words*. The distribution seems to continue beyond the borders of the domain due to the gaussian kernel.

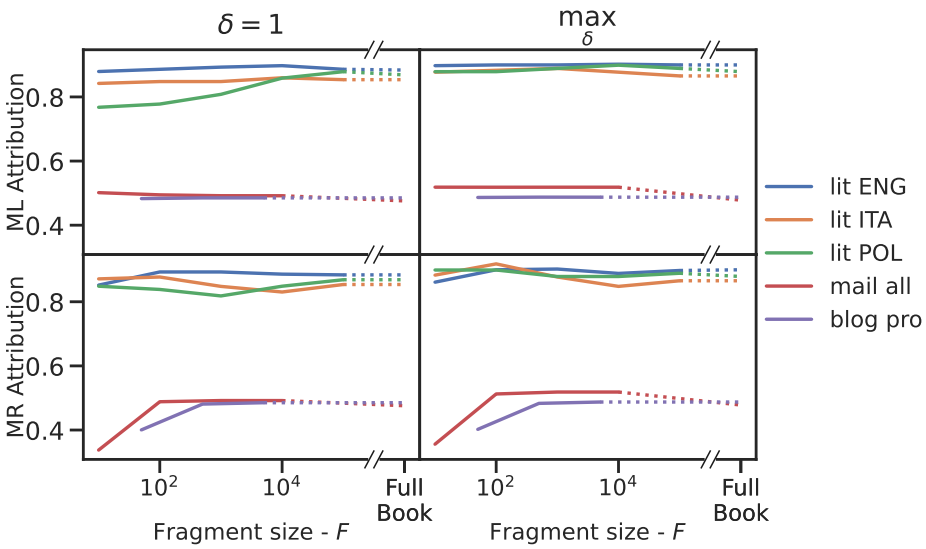


Figure 4.5. Text attribution using *Dictionary words* varying the fragment size. For the informal corpora, a fragment size of 10^4 is already almost equivalent to the use of full texts.

1. `" penguin"` and `"penguin."`
2. `" penguin"`, `"penguins"` and `"enguins!"`

Notice the leading space in `" penguin"`. Here, even if the two texts do not share any word, the use of 8-grams (or shorter ones) allows identifying both texts as involving penguins. This is obtained without an actual stemming phase and is thus less dependent on the language. This selection of longer words corresponds to a culling of the most frequent words in most languages. On the other hand, characters in longer, less common words may appear more than once. If less common words are more concentrated on a few authors, see figure 4.3, we are weighting more the words that may better distinguish the author. In our example, we still had a single shared token if we retained all the N-grams.

The choice of N, however, demands some care. With small values of N, we soon observe all the possible – or admissible for a specific language – N-grams. This does not imply saturation as a large number of N-grams like `"!=."` or `".,:"`, in the case of *3*-grams, continue to appear. The origin of these N-grams may vary, from typographic conventions in literary corpora to emphatic symbols in informal corpora (e.g. sequences of random symbols used to replace profanity).

Using large values of N presents a different set of problems. The first is in terms of information loss. In the example above, using 8-grams, the words `I`, `love`, `my`, `Look`, `at` and `the` simply disappear from the corpus leaving no trace. The second problem is that while the total observed N-grams drop, the number of possible ones grows exponentially[1]. This leads to an expanding set of tokens that grows almost linearly scanning the corpus. The token frequencies tend to be those of a degenerate PD process. Indeed, the number of possible N-grams snowballs but the total number of tokens decreases. For N high enough, the number of tokens is so depressed that we observe fewer different tokens. Most tokens appear only once or very few times leading to unstable probabilities.

Figure 4.6 shows this initial growth followed by a reduction. We shall call N^* the value of N that gives the maximum number of different tokens in each corpus. This value will be a reference when analysing the characteristics of the extracted tokens.

[1]The number of N-grams allowed in any language grows at a much slower pace, and at some point even decrease, most European languages have very few 20-letters words. However, the tendency of many languages to agglutination (e.g. in Turkish as in "Çekoslovakyalılaştıramadıklarımızdanmışsınız", i.e. "You are one of those that we were not able to convert into Czechoslovakians", said of someone who does not change and sticks out in a group) or the use of compound words (e.g. the "Cattle marking and beef labelling supervision duties delegation law" approved in 1999 in Mecklenburg-Vorpommern was titled "Rinderkennzeichnungs- und Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz") clearly show how this limit can often be pushed quite far so that a general rule is unfeasible.

The average number of tokens' occurrences drops from about 10^3 to less than 10 for $N \in [3, 12]$. In informal corpora, we also have a significant fraction of empty texts.

In this varied landscape, we risk losing relevant information but we can draw some helpful insight from the role of punctuation. In this case, we call the complete set of non-alphanumeric characters by the name punctuation. The presence of non-alphabetical symbols is higher than in the raw texts, as words surrounded by punctuation are longer. We may be interested in limiting its impact on our analyses if it does not contain valuable information on authorship for our model. In figure 4.7 we present the percentage of symbols that are not letters or numbers over the total characters of extracted tokens. The higher fractions for the corpora in English, both literary and informal, are probably due to shorter words compared to Italian and Polish. With fewer N-letters words, there is a more significant contribution of $(N - l)$-letters words followed (or preceded) by l punctuation symbols.

As already noted, short N-grams contain much punctuation. We thus have to take punctuation under consideration. Punctuation and other non-alphabetical characters follow less strict rules than usual orthography. For example, a misplaced comma is considered a slight error, and often also experts disagree on where is the right place to put one. This freedom of use is even more evident in informal texts. Many symbols at the end of a word or standing alone may introduce many different tokens. This mechanism leads to many tokens containing a lot of punctuation.

A third quantity we can observe is the number of *hapax* and *dis legomena*. These are tokens appearing respectively once and twice in a text. The analysis of these classes of tokens may provide interesting information. Our approach can extract information from the occurrence of repeated tokens. A text or a fragment whose tokens appear only once would change the nature of our approach, possibly limiting its functionality.

If we look at the fraction of *dis* and *hapax legomena* in texts, see figure 4.8, we notice a different behaviour for literary and informal corpora. The number of *hapaxes* increases quickly with N. The fraction becomes greater than the one observed with *Dictionary words* already for $N \approx 5$ for informal corpora and $N \approx 8$ for literary corpora. The fraction of *dis legomena* in literary and informal corpora is slightly larger than with *Dictionary words*, for $N = 8$ and 5. However, this increase is at the expense of tokens recurring more often. Having repeated tokens is useful for attribution[2]. This is another hint suggesting to keep N small when working on informal corpora. These trends imply a more decisive role of author- and text-specific tokens using this method compared to *Dictionary words*.

[2]Of course, we need repeated tokens in the same fragment. This is likely when working with short texts or long fragments, as in the case of the informal corpora. For literary texts, we can make no assumptions without knowing the clustering of N-grams.

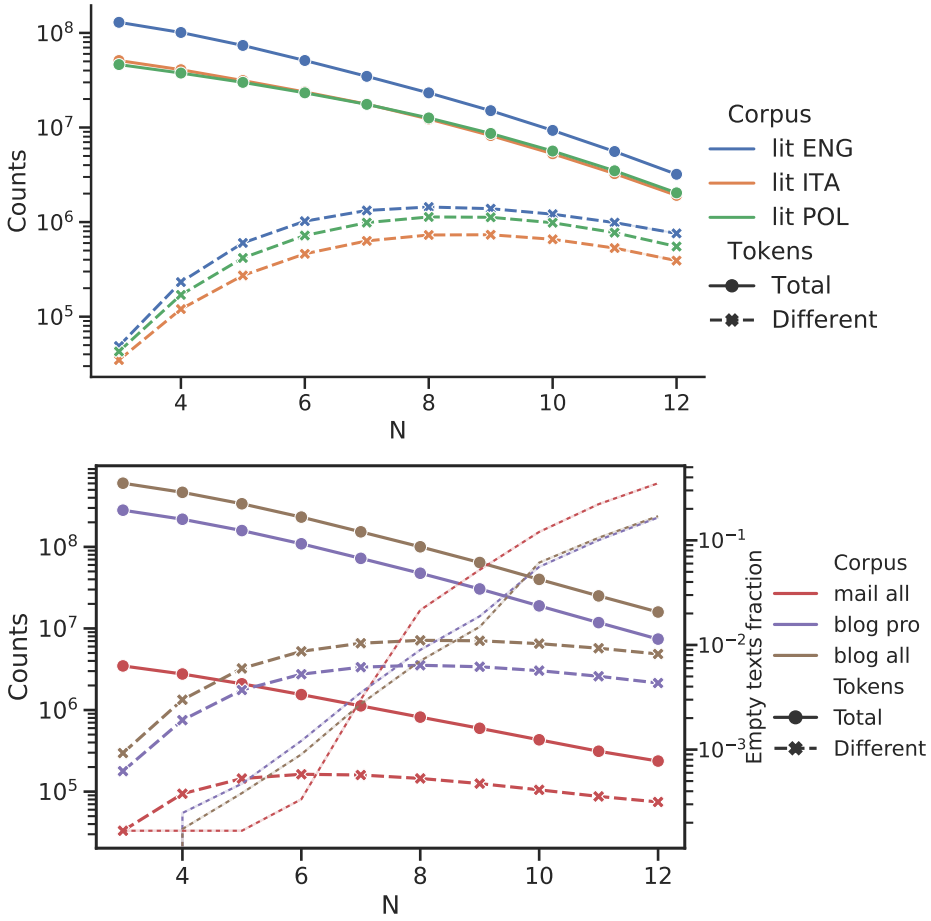


Figure 4.6. Number of tokens and of unique tokens in the various corpora varying the length of the *N*-grams. The top panel depicts the literary corpora, the bottom panel the informal corpora. Some texts are very short in the informal corpora and may contain no tokens if N is too large. On the right of the bottom panel is reported the scale with the fraction of texts with no tokens (dotted lines).

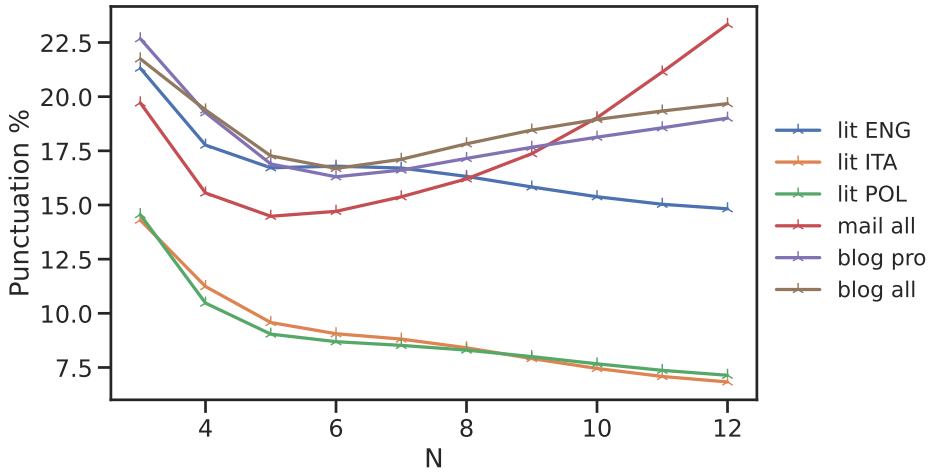


Figure 4.7. Fraction of punctuation characters varying N. In raw texts the percentage of symbols is always smaller. It is 3.72% for the literary English corpus, 3.18% for literary Italian, 4.05% for literary Polish, 3.21% for the Email corpus, 3.99% and 3.81% for the Blog corpus considering all authors and the prolific ones respectively. For the informal corpora the share of punctuation has a minimum for $N \leq N^*$.

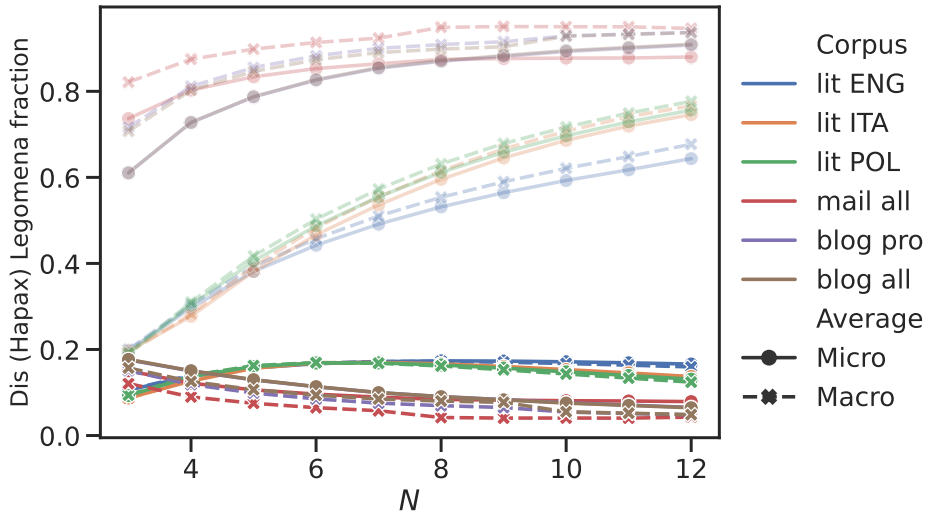


Figure 4.8. Fraction of *dis legomena* using *OSF N-grams*. Shaded in the background the fraction of *hapaxes*.

When using N-grams, the set of tokens of every author changes with N. In figure 4.9 we report the average value of the parameters for all the authors of each corpus. Literary and informal corpora are well separated for both parameters for most of the values of N. However, a global trend is evident: The average value of α tends to grow for every corpus while θ shows a maximum and then tends to decrease again. Also for N-grams, for values of N not too high, the optimisation converges for all authors and corpora. Only in a few cases, the optimisation fails for large N.

Without other factors pushing towards different choices of N (e.g. uneven distribution of word lengths leading to almost empty texts), we systematically look for the value that maximises the results on the training corpus. For the moment, we consider all the options for P_0 and present only the best result. Regarding fragment lengths instead, we present curves for fragment from 10 to 200K tokens spaced roughly a third of decade, i.e. $F = c \times 10^l$ with $c \in [1, 2, 5]$ and $l = 1, 2, \ldots$. This allows showing the effect of the parameter δ introduced in Eq. (3.2). For each corpus, we present the results without the parameter δ, i.e. fixing $\delta = 1$, and the maximum obtained varying δ.

In figures 4.10 and 4.11 we report the attribution scores for the literary English and Italian corpora. Results for the Email and Blog – prolific authors corpora are in figures 4.12 and 4.13. Results for the literary Polish corpus are in Appendix C, figure C.3. We present the results using attribution computed with Maximum Likelihood estimation and Majority Rule to compare their power. We report the best scores for every corpus in table C.2.

The first observation from fig. 4.10 is that the curves in the right panels seem to collapse on the maximum value. The use of δ levels out the differences between fragments' lengths. There is still a noticeable difference despite δ only when using full texts (no fragments at all) or short fragments and MR attribution.

A second observation is that the N coordinate of the maximum increases when using shorter fragments. As the δ parameter compensates well this effect, we may suppose that it is due to the growth with N of the possible words[3].

In the case of the Email corpus, we immediately notice the dramatic drop in attribution between $N = 8$ and $N = 9$. This drop is not reduced by any means by

[3]We get good results with many different tokens and short fragments. In this case, we expect relatively small multiplicities for the tokens. One may object that we are not taking advantage of the PD process. In that case, our method would be just counting how many of the tokens in the book were already present in the author's corpus. Trying to reduce our method to this trivial idea, even considering P_0, did not give results better than 20% of attributions on literary corpora. If we add back the information about the token frequency in the reference author but assuming a simpler Yule-Simon model 1.1.1, we obtain 10 to 50% worse results.

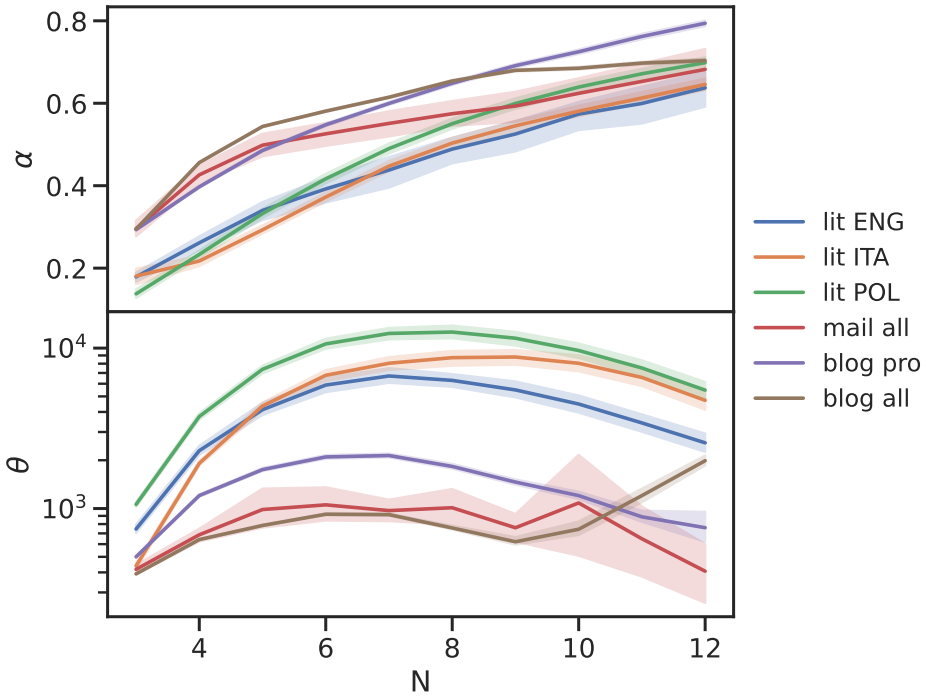


Figure 4.9. Average and 95% confidence interval of α and θ using *N*-grams and varying N. For every corpus there's a (relative) maximum in θ for $N = N^* \pm 1$. Points where the optimisation did not converge, only for high Ns were excluded as their value is arbitrary. In the Email corpus, for $N \geq 9$, there are seven cases (out of 288) of authors with one or zero tokens. In the complete Blog corpus, for $N \geq 7$, there are six cases of authors with no tokens and sixty-five where α did not converge out of 116×10^3.

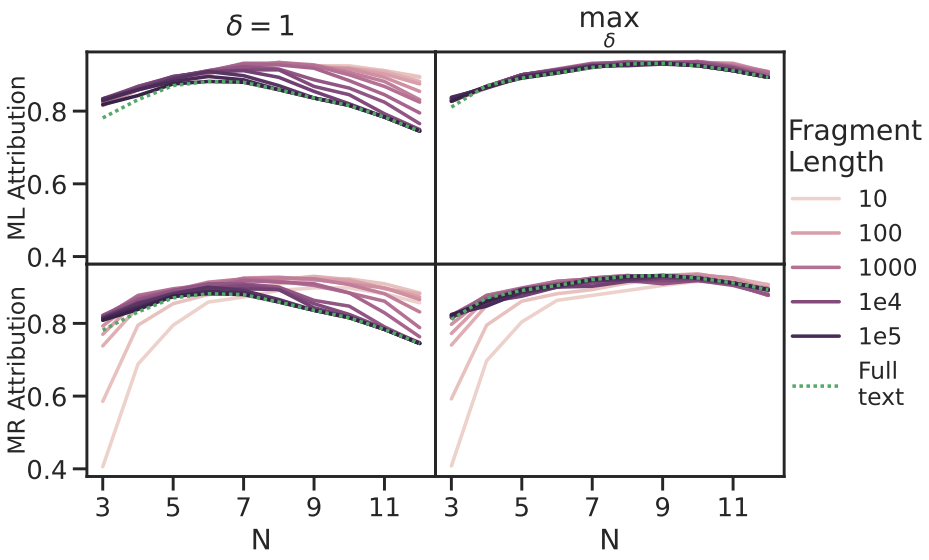


Figure 4.10. Text attribution in the literary English corpus varying N. In the left panels, we notice how the length of the fragments is related to the N-gram size that determines the best score. In the left panels, tuning δ almost completely suppresses the differences between fragment lengths. The maximum attribution scores without δ are 93.4% using Maximum Likelihood and 92.9% using Majority Rule. The use of δ brings the attributions to 93.6% both using ML and MR.

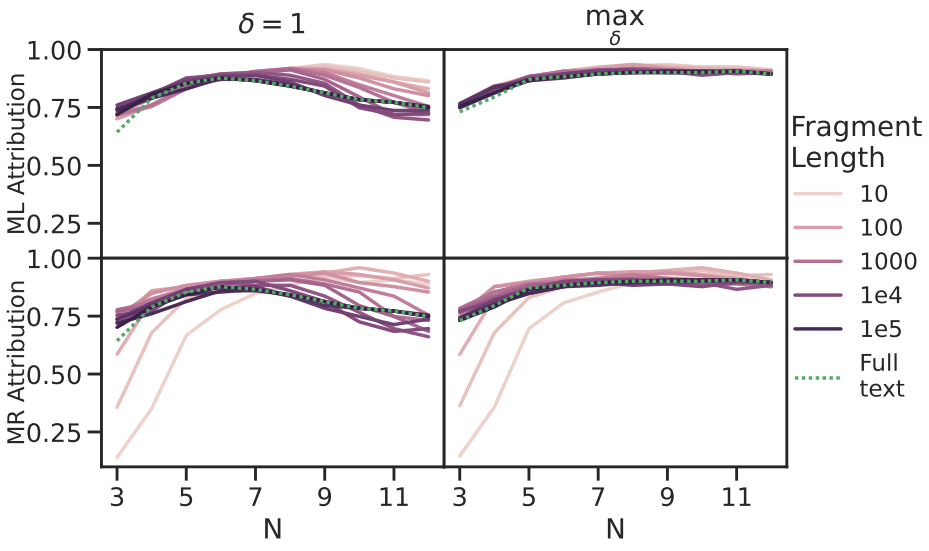


Figure 4.11. Text attribution in the literary Italian corpus varying N. The maximum attribution scores using ML are 93.6% with or without the tuning of δ. The use of MR attribution brings the score to 95.9% both with or without the tuning of δ.

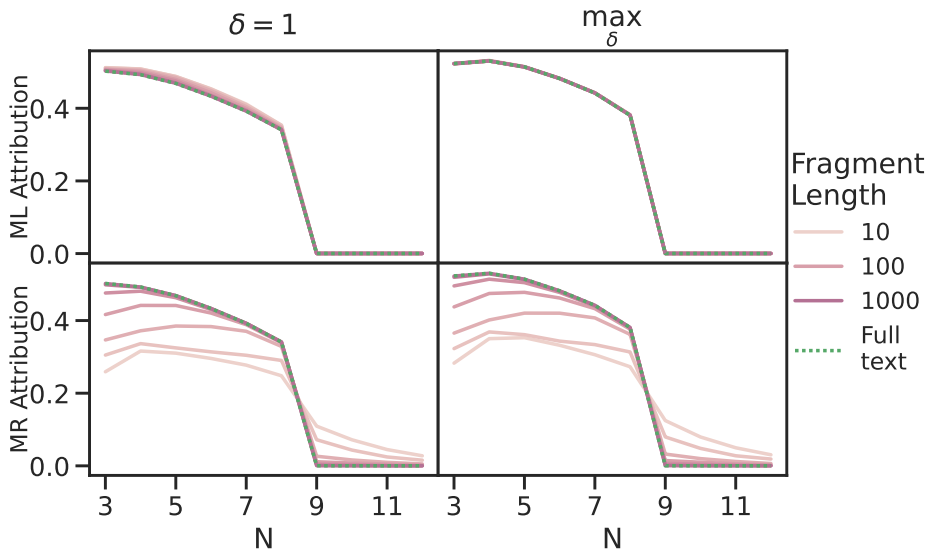


Figure 4.12. Text attribution in the Email corpus varying N. The fragment size was limited to 1000 tokens as already almost equivalent to the use of full texts. The best results with ML and MR attribution are 51.2% and 50.2% with $\delta = 1$, and 53.2% and 53.0% tuning δ.

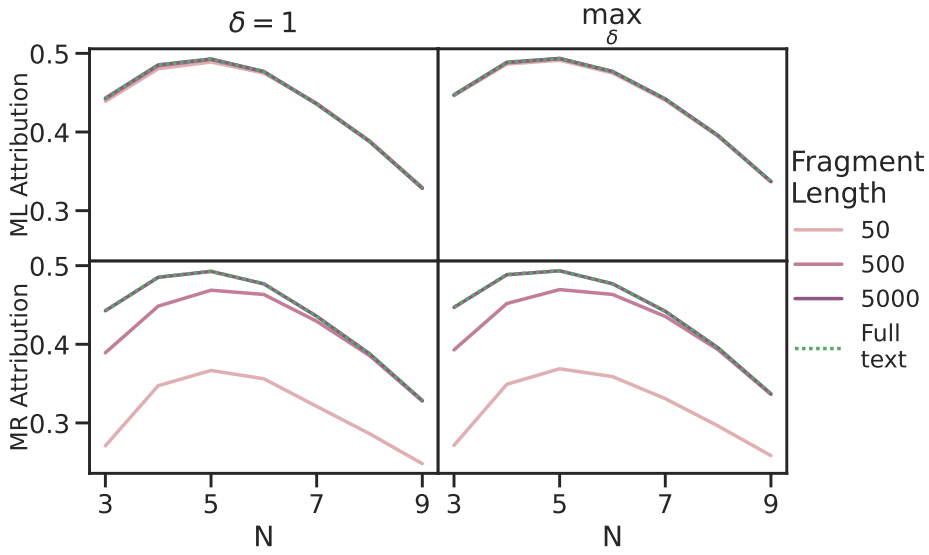


Figure 4.13. Text attribution in the Blog corpus – prolific authors – varying N. Due to the corpus size, we limited our analysis to three sizes of fragments plus full texts. The effect of the tuning of δ is almost unnoticeable. The size of the fragments strongly affects the MR attribution up to fragments size beyond the median post size. With $\delta = 1$ the correctly identified authors are 49.3% using ML and MR attribution. Tuning δ the results improve by 0.1%.

tuning δ but is made smoother by the MR attribution. Also in this case, the tuning of δ cancels the differences between fragment lengths when using ML attribution.

Looking at figures 4.10 and 4.11, we notice that for literary corpora[4], the best scores are for N within $N^* \pm 1$. That is that value that allows a great variety of N-grams whose observation is not yet depressed by the falling number of total N-grams. In these conditions, the set of possible N-grams behave as effectively infinite, and the number of counts for common and rare N-grams still differs.

For informal corpora, figures 4.12 and 4.13, we find the best results for values of N smaller than N^*. For the Email corpus, the best results are for $N = 4$ while $N^* = 6$. For the prolific authors' subset of the Blog corpus, the best results are for $N = 5$ while $N^* = 8$. Indeed, we notice that, for the informal corpora, maximising the number of different N-grams takes us in a region where the fraction of hapaxes is ~ 0.8, and we have a non-negligible fraction of empty documents.

Interestingly, the best scores are for values of N close to the minimum fraction of punctuation. This could be a guidance for these kinds of corpora. Notice that literary corpora have no global minimum in the interval of values considered.

Possibly, an even better indicator of the region of N that gives the best scores is θ. For all corpora, the best results seem to be in the immediate neighbourhood of the first maximum of the average value of the concentration parameter.

These kinds of observations can guide the search for the optimal value of N in cases where – due to the size of the corpus or limited resources – a full search is not feasible. Note that the trend of the score varying N is smooth, and other standard procedures for finding the maximum should work well. In the following, we will present the results for each corpus only for the N that maximises attribution. Graphs for all the values of N in consideration can be found in Appendix C, figures C.1 and C.2.

In table 4.3 we report the fraction of characters in the original text that is included in the tokenised sequence. Any character may appear more than one time, if it finds itself in the overlap of multiple N-grams. The fraction reported refers to the characters appearing *at least* once in the tokenised sequence. We notice that for corpora with a smaller optimal N, informal corpora, the fraction of characters retained after tokenisation is higher than in the literary ones. Therefore, we suggest that – choosing N – we select the subset best represented by a PD process even if we lose some information.

Also in the case of *OSF N-grams*, the frequency rank of the tokens in the corpus shows a power-law tail, see Fig. 4.14a. The same is true also for Heaps' law, see Fig. 4.14b. Again we refer to a sequence obtained by random concatenation of all

[4]Similar results in figure C.3 for the Polish corpus.

Table 4.3. Characters appearing at least once in the tokenised sequence for every character in the original text. Values for the N that maximises attribution. The micro average is computed on the whole corpus at once, the macro average is the average over the per text fraction of characters preserved.

Corpus		Average	
		Micro	Macro
Literary	Polish	0.416	0.421
	Italian	0.470	0.468
	English	0.458	0.456
Email		0.752	0.732
Blog	all authors	0.730	0.737
	prolific	0.733	0.738

the texts in a corpus. We notice that the curves for the different corpora, given the different N, show similar behaviours. The heaps' exponents are higher than in the *Dictionary words* case but more dispersed.

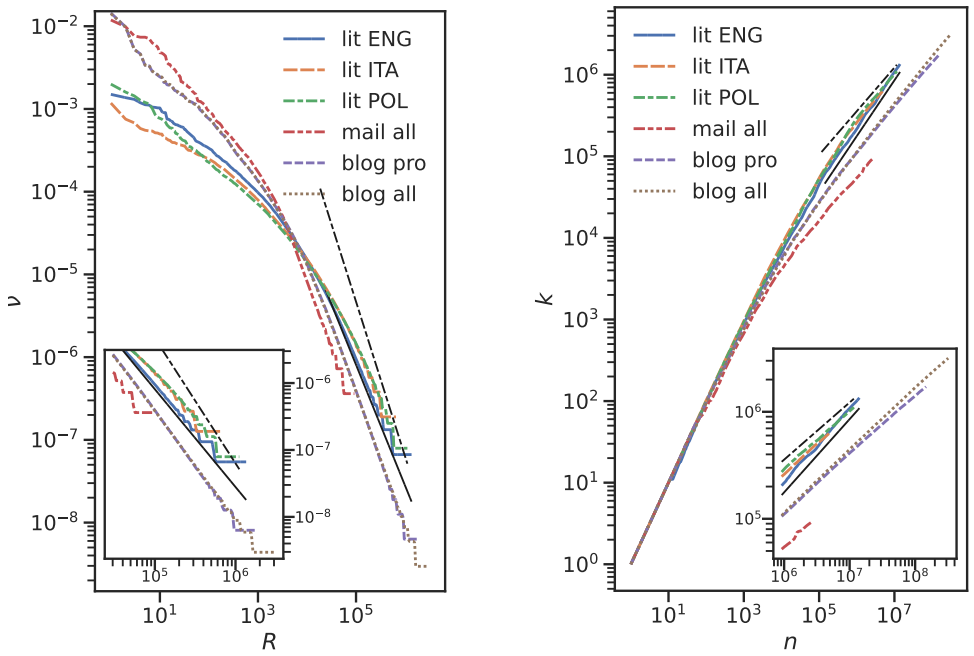
Also, in the case of N-grams, all corpora exhibit Taylor's law. In figure 4.15, we observe that the curves for all corpora grow paired for a first tract. For higher values of k, the curves for informal corpora detach and grow slightly faster.

Finally, we show in Fig. 4.16 the fluctuations of the tokens' frequency across different authors. As in the case of *Dictionary words*, we notice that as the fluctuations decrease with the frequency (higher ranks), their relative amplitude grows.

4.3 LZ77 Sequences

The last kind of variables considered are *LZ77 sequences* obtained from a compression algorithm to capture repeated strings from word to sentence level. We used an implementation of the LZ77 algorithm derived from the one in the *gzip* software [38]. The algorithm records only repeated sequences of at least 3 bytes, and the tokens may include punctuation and spaces in any number.

The *LZ77* algorithm searches backwards on the sequence for an earlier occurrence of the following characters. This behaviour is needed to compress a file in a single pass as there is no dependency on future characters. If a stationary ergodic source produces the sequence we are compressing, the asymptotic compression rate approaches the entropy rate of the source [155]. However, the algorithm limits how far back to look for an earlier occurrence. To optimise space and time, there is a sliding window on the sequence where to look for repetitions. Even on a very long text, usual



(a) Zipf's law **(b)** Heaps' law

Figure 4.14. Zipf's law and Heaps' law for the different corpora using *OSF N-grams*. Straight lines in the Heaps' law plots show functions of the form $f(x) = ax^{\beta}$, with the highest and lowest fitted exponents β equal respectively to $\beta = 0.682$ (literary English) and $\beta = 0.535$ (literary Polish). Other values are $\beta = 0.573$ (literary Italian), $\beta = 0.592$ (Blog – prolific), $\beta = 0.620$ (Blog – all authors), and $\beta = 0.627$ (Email). Straight lines in the Zipf's law plots show functions of the form $f(x) = ax^{-\alpha}$, where the exponent α is equal to β^{-1} for the highest and lowest βs considered above. The inset shows the points closer to the asymptotic limit.

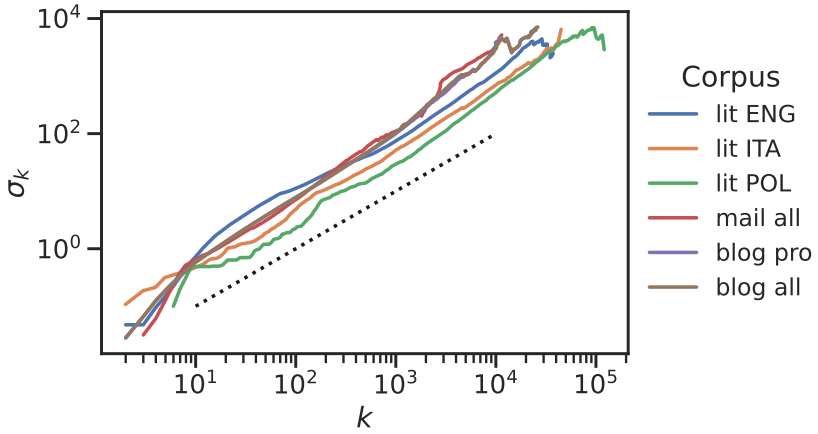


Figure 4.15. Taylor's law using *OSF N-grams*. The dotted line is a power law with exponent one provided as a guide for the eye. The decreasing tract at the end of some curves is due to the reduced number of texts reaching very high values of k.

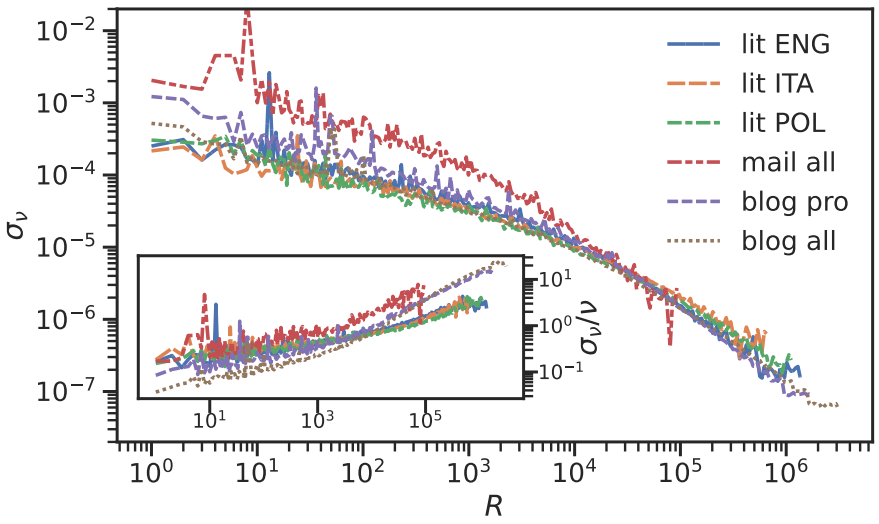


Figure 4.16. Fluctuations in token frequency across authors using *OSF N-grams*. The main graph shows the standard deviation across authors of the frequency of tokens, ordered by global frequency. The inset shows the growth with the rank of the relative amplitude of the deviation.

implementations of the algorithm will not search more than 2^{15} characters back, roughly 4 KB.

More in detail let $x = x_1, \ldots, x_N$ be the sequence to compress, where x_i represents a character in the alphabet, in our case a byte. The LZ77 algorithm replaces the second occurrence of a string with a reference to the previous string defined by the distance, how far back into the window the sequence starts, and the length of the repeated string, see Fig. 4.17a. As the algorithm scans the sequence, when the first n characters have been codified, it looks for the largest integer m such that the string $x_{n+1}, \ldots, x_{n+m}$ already appeared in $x_1, \ldots, x_n$. Then it replaces the next m characters with two numbers: the distance between the two strings and the length m of the match. If the algorithm does not find any match, then it codifies the next character, x_{n+1}, as itself. Most of the unmatched text happens when codifying the first characters of the sequence, leaving some characters uncompressed, but becomes very infrequent as the procedure goes on.

We can easily adapt this algorithm to our needs. First, one wants to avoid biases in sampling the head of the sequence. The original algorithm searches a very short sequence during the first steps and finds a few matches. This produces a few tokens for the first part. In our case, since we are not interested in the compression itself,

we can accept a match in the "future" of the sequence. We treat the sequence as a loop, so – in the first steps – we search for matches in the last part of the sequence[5].

We can illustrate the different behaviours with an example, see Fig. 4.17b. In the right panel, the last characters of the sequence appear again before the beginning. This allows matching the first occurrence of `asdQW` with the second. Of the 17 characters of the string, the original version in the left panel matches only 8 towards the end. In the adapted version, we match 13 characters with no bias.

The second edit to the algorithm is in the size of the window. The choice of the window presents problems similar to those observed with the different values of N in the *OSF N-grams* tokenisation. The use of long windows allows finding matches for most of the sequence. However, these matches can be very long and specific to the text itself. In this case, they will recur a few other times in the corpus: few, rare tokens. On the other hand, a small window reduces the chances of a long match – or a match *tout court* – so that there are fewer different tokens, relatively short and leaving out large portions of the sequence. Also in this case, it is fundamental to find the right balance.

When using *LZ77 sequences*, we again consider all the non-alphabetical characters. However, this time we keep only sequences of characters that appear at least two times in the sliding window.

We shall decide the length of the compressor window. In figure 4.18 we show how the number of total tokens and the number of different tokens varies with the window length. As the window length grows, the probability of a match increases, as does the average length of the matched sequences, see figure 4.19. Long matches imply fewer tokens overall but more different ones.

In this case, the number of different tokens is not a helpful indicator, and other problems arise. Some short texts have no repeated sequences. This is not observed in texts longer than 157 characters, but they can represent a relevant fraction in specific corpora. For the Blog corpus, both with all the authors or only the most prolific thousand, the fraction of empty texts is significant but not above 4%. In the case of the Email corpus, 9% of the texts have no repeated sequences. These fractions are constant across the different window lengths as all the texts with no repetitions have less than 158 characters, and about 80% of them has less than 50. This difference is mainly due to the conciseness typical to the email medium compared to the more narrative style of blog posts. More than 92% of the texts in the Email corpus are shorter than the median blog post.

[5]Of course, if the sequence is shorter than the window, we take care to avoid matches with the same section.

(a) Original version of the algorithm

Original sequence

asdQWEjlasdQWkasd

Compressed sequence

asdQWEjl(8,5)k(15,3)

Repeated strings found

asdQW asd

(b) Adapted version of the algorithm

Original sequence

...jlasdQWkasdasdQWEjlasdQWkasd

Compressed sequence

(9,5)Ejl(8,5)j(15,3)

Repeated strings found

asdQW asdQW asd

Figure 4.17. Demonstration of the LZ77 algorithm.

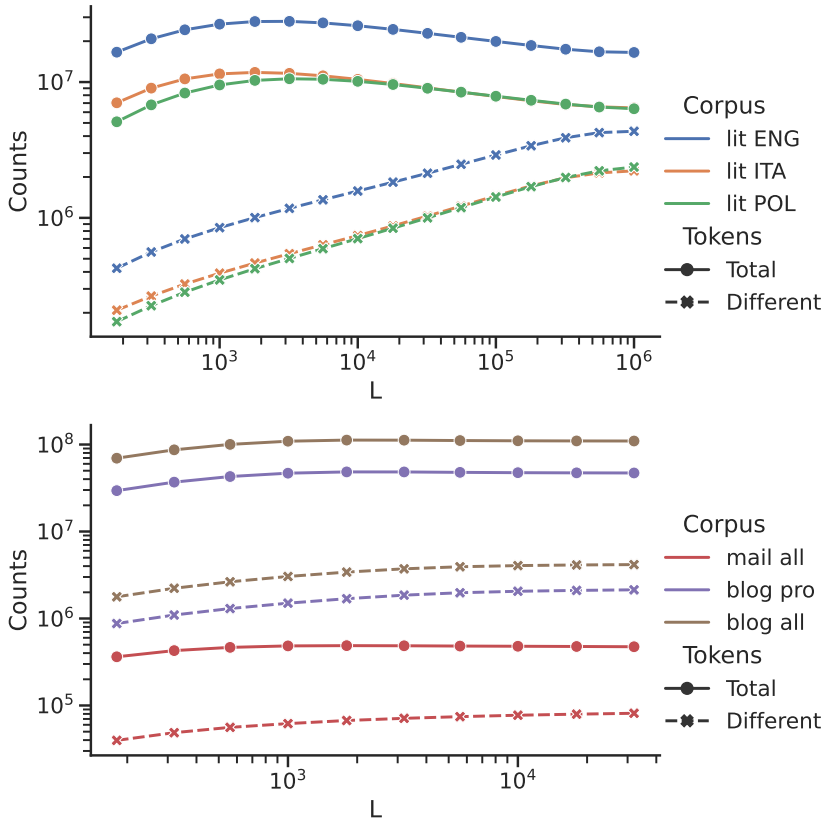


Figure 4.18. Number of tokens and unique tokens in the various corpora varying the length of the window. The top panel depicts the literary corpora, the bottom panel the Email and the Blog corpora. The window sizes considered here are in the range $[180, 32000]$ characters for the informal corpora and $[180, 560000]$ for the literary ones. Points are spanned a quarter of decade, i.e. for $L = w \times 10^l$ with $w \in [10, 18, 32, 56]$ and $l = 1, 2, \ldots$.

These considerations on the lengths of the texts take us to another consideration. The length of the text limits the size of the *LZ77* window. When some texts are shorter than the window, the preprocessed corpus contains a mixture of different effective window sizes. These are unwanted differences as the tokens extracted for different sizes of the window have different properties – for example, the token length distribution, see figure 4.19 – and cannot be compared directly. In the case of literary corpora, books are hundreds of thousands of characters long, and this problem arises only with window lengths approaching the millions of characters. In the case of informal texts, this plays an important role. For example, one-fifth of the texts in the Blog corpus and more than one half in the Email corpus are shorter than the shortest window considered, 180 characters. This is the cause of the flatness of the curves for the informal corpora in figures 4.18 and 4.20: the effective window size ceases to increase.

We conclude that the *LZ77 sequences* tokenisation is not suitable for corpora with concise texts. Therefore, in the following, we consider only literary corpora when dealing with *LZ77 sequences*.

In figure 4.20, we report the fraction of characters in the original text that appears in the tokenised sequence. The differences between the micro and macro averages in the informal corpora tell that most of the preserved characters are in long texts. Short texts lose most of their content. As the length of the window grows, the algorithm preserves more and more characters as it is easier to find a match anywhere else in the window. This is obtained at the price of more specific tokens where the frequency of the most common ones drops two orders of magnitude, see Fig. 4.21a.

Also in this case, the tail of the frequency rank of the tokens in the corpus is in agreement with a power-law, see Fig. 4.21a. The same is true also for Heaps' law, see Fig. 4.21b. Again we refer to a sequence obtained by random concatenation of all the texts in a corpus. The limit exponent changes slowly, varying the window size, less than 0.1 over four decades of window length.

The fraction of *dis* and *hapax legomena*, see figure 4.22 has an interesting behaviour. Because of the definition of *LZ77 sequences*, each text containing at least one token usually contains two equal tokens. This is true unless one of the two repetitions falls outside the window. Remind here that we treat our texts as rings, and – to have this phenomenon – we need two repetitions close to each other and a text longer than the window. One may thus expect a greater amount of *hapax legomena* when using short windows. Many of the repetitions will fall outside. With longer windows, the number of *dis legomena* and tokens appearing more times should increase.

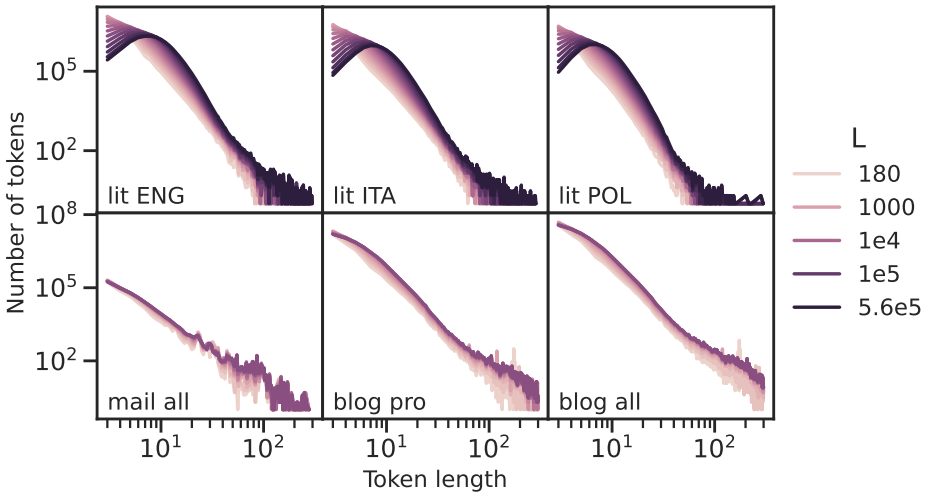


Figure 4.19. Token length distribution in the various corpora varying the length of the window. The top panels depict the literary corpora, and the bottom the informal. Increasing the size of the window, the distribution of the token lengths shifts to the right. In the informal corpora, the presence of short texts limits the effective increase of the window. Because of this, the number of short tokens does not decrease significantly, increasing the window size.

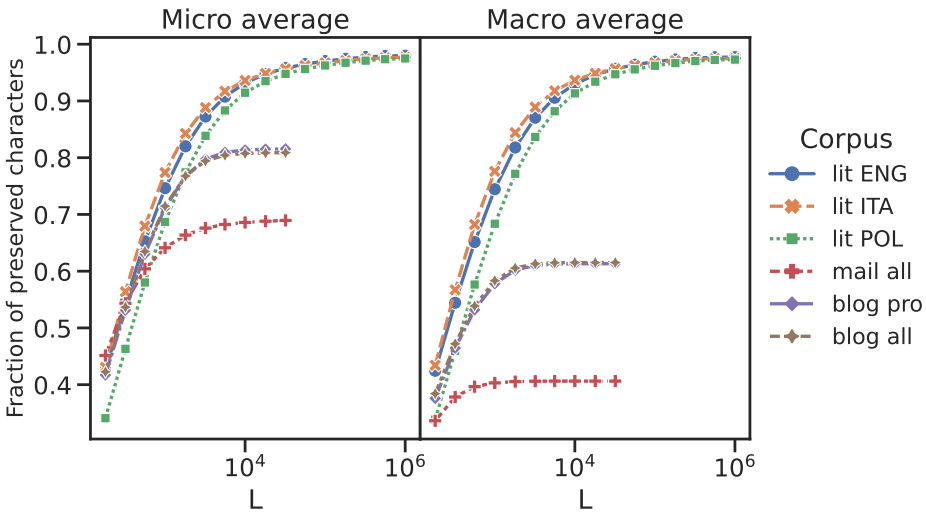
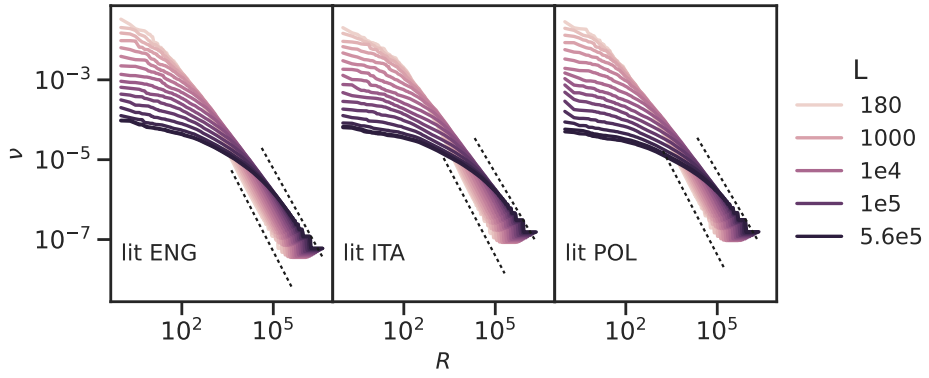
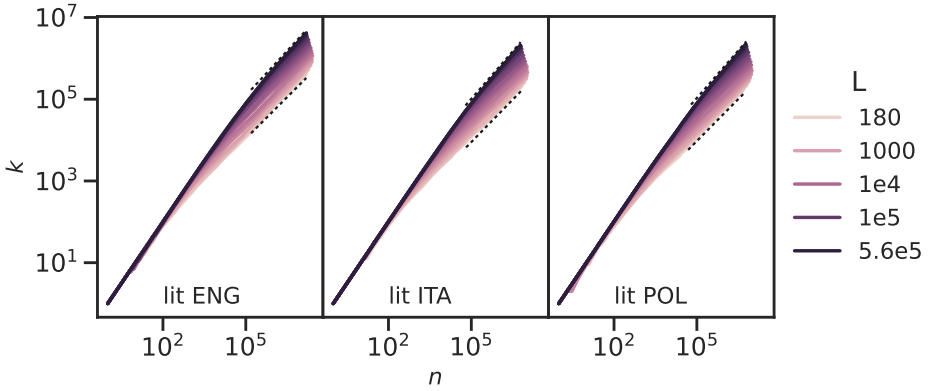


Figure 4.20. Fraction of characters of the original text preserved through tokenisation varying the length of the window. With the literary corpora – increasing the window size – more and more sequences are found to have a replica. In informal texts, the effective window length is limited by text length. When the window is longer than the great majority of texts, increasing its size will have no effect.



(a) Zipf's law



(b) Heaps' law

Figure 4.21. Zipf's law and Heaps' law for the three literary corpora using *LZ77 sequences*. Straight lines in the Heaps' law plots show functions of the form $f(x) = ax^{\beta}$ for window lengths of 180 and 560000 characters. The fitted exponent β equals to $\beta_{180} = 0.677$ and $\beta_{560000} = 0.730$ (literary English), $\beta_{180} = 0.692$ and $\beta_{560000} = 0.775$ (literary Italian), $\beta_{180} = 0.684$ and $\beta_{560000} = 0.773$ (literary Polish). Straight lines in the Zipf's law plots show functions of the form $f(x) = ax^{-\alpha}$, where the exponent α is equal to β^{-1} for the different βs considered above. Increasing the window size, the frequency of low-rank tokens decrease in favour of high-rank ones. This growth causes a longer tract with exponent ~ 1 in the Heaps law graph. The above graphs for the informal corpora are in figures C.4a and C.4b.

Indeed the number of *dis legomena* grows while the *hapaxes* follow a more complex pattern. As the window size grows, it is easy to find substrings of other matches, and – as the average match length increases – there are more possible substrings, possibly appearing only once. The number of *hapaxes* depends on the ease of finding such substrings. This property is corpus dependent, and the Polish corpus has the highest fraction of *hapaxes*.

For windows shorter than 10^4 the fraction of *hapaxes* decreases without a significant increase in *dis legomena*. Many tokens occur more than twice. With longer windows the number of *hapaxes* and *dis legomena* increases. This behaviour is in line with the observed flattening of the frequency-rank plot as the window length increases. The variety of tokens increases as their mean occurrences decreases. If these tokens are specific to the text, we will have poor results dominated by the casual identity of tokens across books. If these tokens are specific to the author, we extract only the most relevant information.

Finally, we show in Fig. 4.23 the fluctuations of the tokens' frequency across different authors. The growth with the rank of the fluctuations' relative amplitude strongly depends on the window size. For short windows, we observe a behaviour similar to other features. The relative fluctuations grow roughly by two orders of magnitude from low to high ranks. On the other hand, the relative fluctuation changes only about one order of magnitude using long windows. As a result, low-rank tokens are less frequent and become more author-specific, while high-rank tokens become more widespread.

Values of $\frac{\sigma_\nu}{\nu} > \sqrt{\#\text{ authors}}$ are allowed if a token is used only by a short author. We computed σ_ν on the macro averaged frequency (average per author frequencies). The author frequency of a token is related to the reinforcement term in the PD process, and we will use it in chapter 6. On the other hand, we divided by the micro average ν (overall frequency) as this is the value used for P_0 in chapter 5. A strongly author-specific token with a lower overall frequency will substantially impact the likelihood of a different author and is reflected by higher values in figure 4.23.

We observe in figure 4.24 very similar behaviours across the languages for the parameters of the PD process for literary corpora. Only the English corpus shows a broader variance. Looking at the parameters, as looking at dictionaries, we see no tipping point. The parameters evolve smoothly for small window sizes L and become more irregular and dispersed for larger windows. Parameters stop changing when the window is longer than most texts. Also for *LZ77 sequences*, considering the authors with all their books, the optimisation converges for every author in the literary corpora.

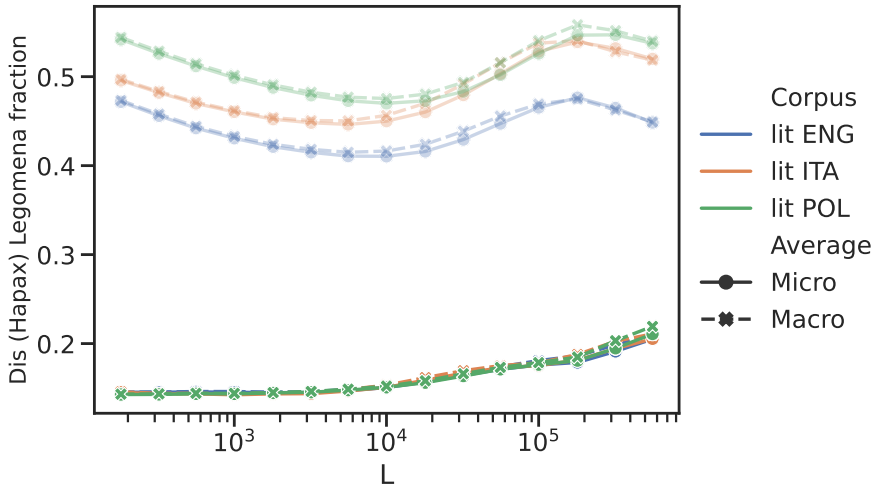


Figure 4.22. Fraction of *dis legomena* using *LZ77 sequences*. Shaded in the background the fraction of *hapaxes*.

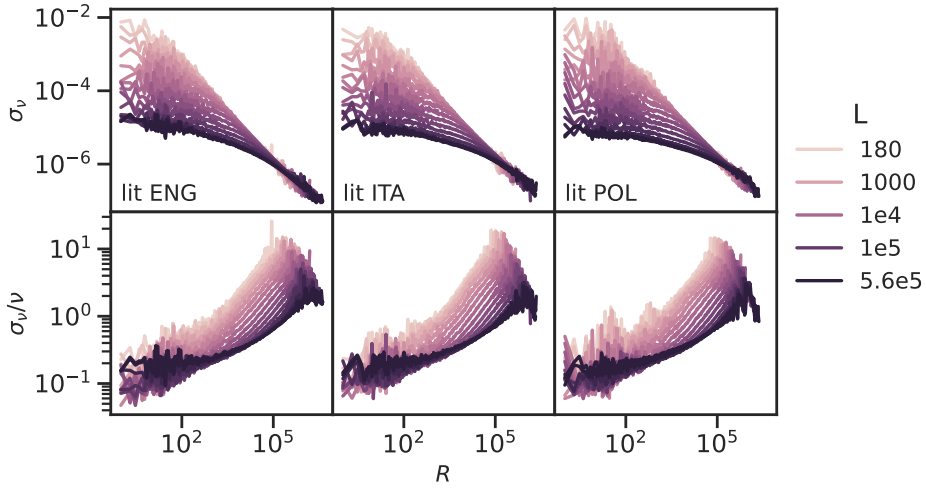


Figure 4.23. Fluctuations in token frequency across authors using *LZ77 sequences* for the literary corpora. The top panels show the standard deviation of tokens' frequency across authors, ordered by global frequency. In the bottom ones, we show the relative variation.

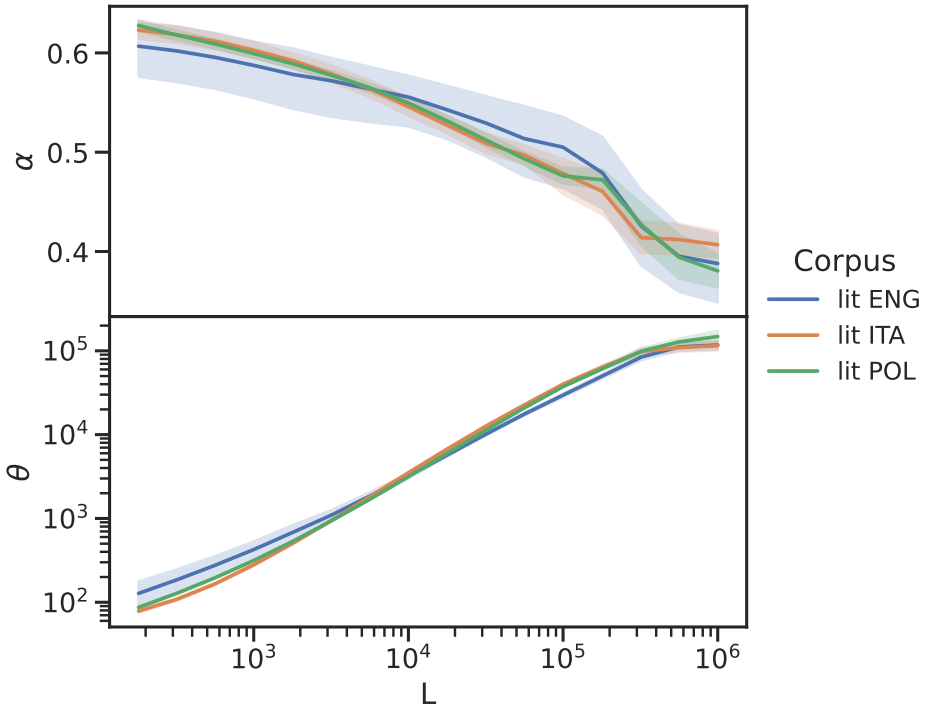


Figure 4.24. Average and 95% confidence interval of α and θ using *LZ77 sequences* and varying the window size. The trends for all corpora are smooth for small windows.

As we did for the *Overlapping Space Free N-grams*, we now present the results changing the free parameter introduced by the tokenisation. In this case, we change the length of the sliding window of the compressor. Using *LZ77 sequences*, the number of different tokens is constantly growing with the window length and it is not a valuable indicator for the best window length.

We have no clear hint on which could be the right size of the window. We observe a general growth of the different tokens number together with the window size. Having many different tokens has proved successful when looking for the correct value of N. However, we also know that we cannot let the window size grow indefinitely. We expect the attribution to improve with the growing different tokens number ut po a window size where texts start to fall short. When texts are shorter than the window, their dictionaries cease to change. Therefore, the dictionaries extracted from short texts will not be directly comparable to those extracted from longer texts.

Looking at the results in figures 4.25 and 4.26 we notice precisely this kind of behaviour. The results for the Polish literary corpus are very similar and presented in figure C.6. We report the best scores for every corpus in table C.3. The number of correctly identified texts grows with the window size L up to large values before the results worsen. In general, there is not a single value of L but a region more or less wide where different fragment lengths show their maximum. We notice that the maximum values for all three corpora occur in the large window regime, in the range where θ is still growing, and the mean α parameter becomes erratic.

Looking at the window sizes that correspond to the maximum values of the attribution, we get another hint on why this features method would not be effective on informal corpora. The best window lengths are not always shorter than the smallest book. We get some of the best results using windows longer than the shortest 20, 29, and 44% of the Italian, Polish, and English corpora books, respectively. In any case, the window is never longer than about ten times the shortest text.

If we were to apply this requirement to the informal corpora, we would be using extremely short windows. Even disregarding the shortest texts, always one or two bytes long, we should not use windows longer than a few hundred characters. As seen above, using shorter windows worsen the results as the compressor finds few repetitions, and most of the text is lost in the processing.

Also for *LZ77 sequences*, the tuning of δ improves the results. The more substantial effect is for longer window sizes. The dramatic fall observed in the left panels almost vanishes when tuning this single parameter.

Figure 4.27 reports Taylor's law curves for the literary corpora. In this case, we used a single window size value from table 4.4.

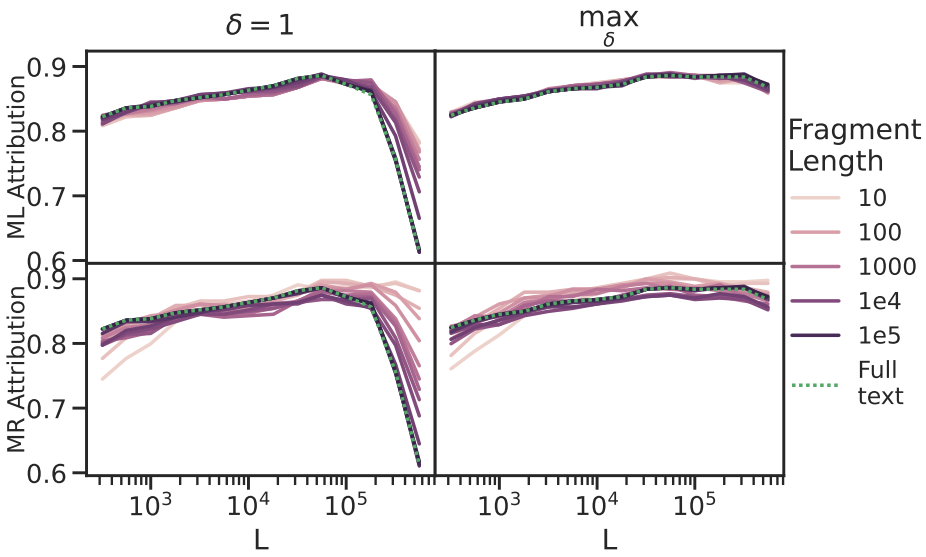


Figure 4.25. Text attribution in the literary English corpus varying the window size. The maximum values are for window lengths ranging from 5.6×10^4 (MR attribution, tuning δ, 90.9% success) to 10^5 (MR attribution, tuning δ, 89.7% success). Using ML attribution the maximum is for $L = 5.6 \times 10^4$, 88.8% success with $\delta = 1$, 89.1% with tuning.

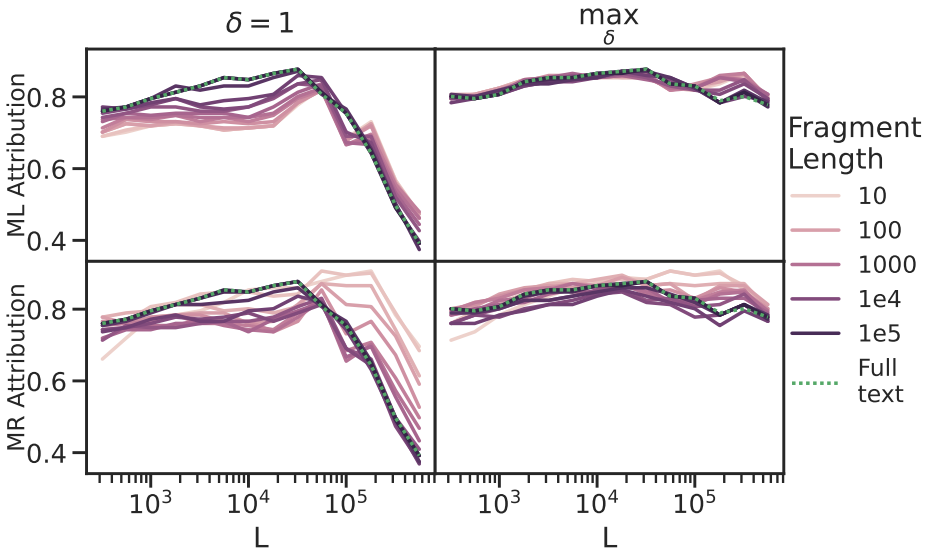


Figure 4.26. Text attribution in the literary Italian corpus varying the window size. The maximum values are for window lengths ranging from 3.2×10^4 (ML attribution, 87.7% success) to 1.8×10^5 (MR attribution, 90.6% success). In both cases the maximum values tuning δ or not are the same.

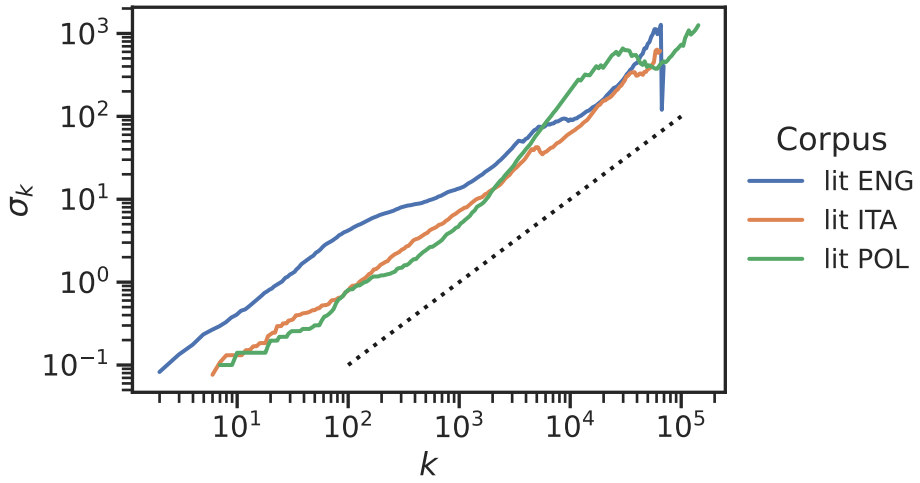


Figure 4.27. Taylor's law using *LZ77 sequences*. The dotted line is a power law with exponent one provided as a guide for the eye. The decreasing tract at the end of some curves is due to the reduced number of texts reaching very high values of k.

4.4 Variable Comparison

It is now time to compare the different preprocessing methods to evaluate if one proves better than others. Since we are now interested in the preprocessing method itself, we will consider the maximum value over all the other parameters. We will keep the fragment length as an independent variable as it will be a central element of the following analyses. Therefore, the definition of P_0, the free parameter of the tokenisation (if present) and δ will all be maxed out.

In figures 4.28 and 4.29 we present the best results for the English and Polish literary corpora while, in figure 4.30, the results for the Email corpus. Results for the literary Italian and Blog–prolific authors corpora are displayed in Appendix C, figures C.7 and C.8 respectively.

Again, we notice that δ reduces the differences due to the length F of the fragments for the literary corpora. In the Email corpus, we observe a general improvement too. However, from these graphs is easier to notice that – even if reduced – a dependency from F remains. For all corpora and attribution methods, except the Email corpora using MR, we have a maximum in the attribution rate for $F \lesssim 100$. This maximum may not be unique as in the case of the Polish literary corpus. Here we obtain the same maximum value using *LZ77 sequences* with fragments of any length, from 10 to 2×10^4 tokens.

This valuable information restricts the range of possible values of F to look for the best attribution.

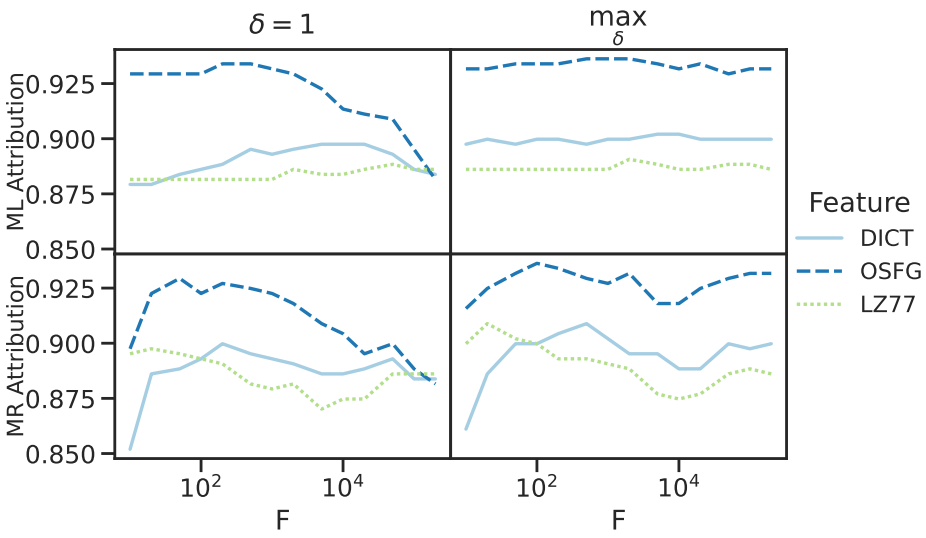


Figure 4.28. Text attribution in the literary English corpus with the three different tokenisation techniques. The minimum value of F considered here is 10, the maximum is 5.6×10^5. The maximum values of attribution are 93.6% when tuning δ both using MR and ML attribution. With $\delta = 1$ the correct attributions are 93.4% and 92.9% using ML and MR respectively.

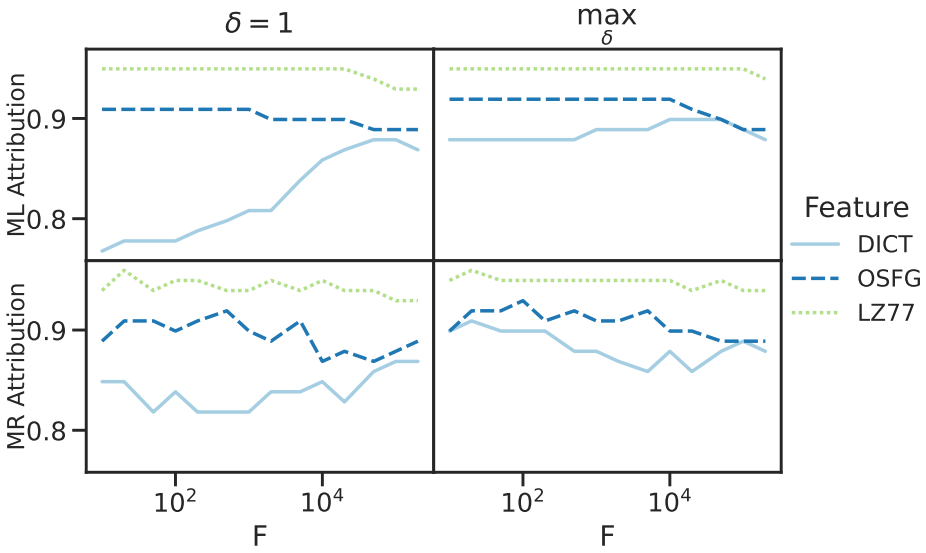


Figure 4.29. Text attribution in the literary Polish corpus with the three different tokenisation techniques The maximum values of attribution are 96.0% using MR and 94.9% using ML. The maximum values are independent from the tuning of δ.

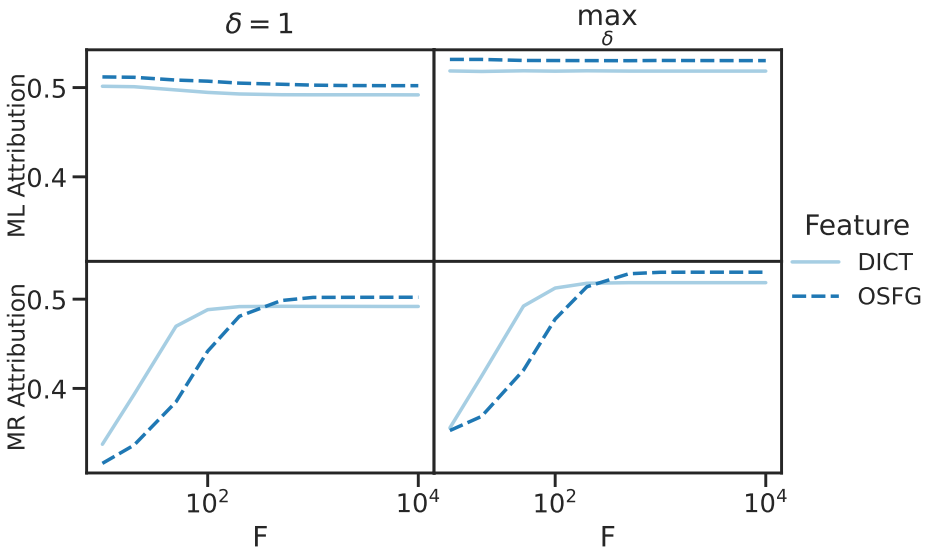


Figure 4.30. Text attribution in the Email corpus with the two different tokenisation techniques. The maximum values of attribution are 53.2% using ML and 53.0% using MR. The maximum values tuning δ, in this case, are $\approx 3\%$ higher than having it fixed to 1. This is a case where we have some very short author sequences. The fifth smallest author has 96 *Dictionary words*, one-fourth of the next bigger one. The fourth-smallest using *4*-grams has 275, again one-fourth of the next one. These are the orders of the fragment length for which the MR attribution keeps up with the ML.

Another interesting result is about the dependency on the language. We designed this method to be language-independent, and indeed we reached 93% to 96% of attribution with all the three languages we considered.

Concerning the best features, for all corpora but literary Polish, the *Overlapping Space Free N-grams* represent the best option. In the case of the Polish corpus, we obtain the best scores using *LZ77 sequences*.

In table 4.4 I summarise the results obtained from the analysis of the different preprocessing methods. In the following, we will use the values reported here, except when differently stated.

Table 4.4. Best features configuration for all corpora. Best values of the free parameter for each corpus and preprocessing method. The values in bold mark the best overall configuration for each corpus. Due to its size the full Blog corpus was excluded from this analysis.The values for the maxima are repeated in tables C.1 to C.3

Corpus		*OSF N-grams*	*LZ77 sequences*
Literary	Polish	10	$\mathbf{3.2 \times 10^5}$
	Italian	**10**	3.2×10^4
	English	**9**	5.6×10^4
Email		**4**	—
Blog	prolific	**5**	—

Chapter 5

Choosing the Base Probability Distribution

We will now discuss the possible choices for the base probability distribution P_0. Every choice leads to a different interpretation of the P_0 and the probability of texts themselves.

The choice is not immediate as our approach lives on an edge. On the one side, the conditional probability uses the form for the continuous base probability distribution. On the other side, the space of the tokens is discrete.

We are committed to keeping the approach simple and functional. However, choosing one of the sides without substantial changes in the approach would compromise the simplicity. Therefore, we decided to take a middle way. We kept the base probability distribution discrete but then posed $t_j = 1 \; \forall j$ in equation (1.36). In this way, we recover the straightforward form of the PD process with a continuous base probability distribution.

Projecting the tokens in a continuous space – possible with tools like LDA or Word2Vec – would be impractical. First, we must estimate the P_0 in a some-hundred-dimensional space. Even the coarsest estimate from data would require wild assumptions. Then, we have the problem of connecting the processes representing different authors. Indeed, using a non-atomic base distribution, the probability that two processes draw the same token would be zero. The most sensible choice would be to resort to a hierarchical model with a discrete PD process to sample on and at least one layer of hierarchy that requires optimisation.

Using a PD process with atomic base probability distribution requires to sample the number of extractions from P_0. Preliminary results on a subset of the Italian literary corpus showed worse results than the CP-DP. We considered all books from the authors with up to four books and used full books (no fragments) and *Dictionary words*. We applied a correction to the probability obtained with the

discrete probability to compensate for the different lengths of the author[1]. The final score is 0.81 compared to 0.94 of the CP-DP with the same hyperparameters. Fixing $\delta = 1$, we still get a score of 0.91.

Even if the discrete form results are not encouraging, they are useful to estimate the magnitude of our approximation. On a sample book, about 80% of the tokens have an estimated $\langle t_j \rangle < 2$. However, the remaining 20% accounts for about 75% of the total extractions from P_0. These are common tokens and have values of t_j reaching the hundreds.

The values of t_j are also an interesting byproduct of the probability estimate. In general, the t_j grow with a power-law like $t_j \propto n_j^\alpha$ with $\alpha < 1$. However, for some tokens, we find an estimate of t_j significantly smaller than expected from the above relationship. For example, in *Ogni promessa* by Andrea Balzani, the words that most differ from the expected values are the main characters' names and Russian places, relatives, and objects that have a key role in the plot.

Once we choose the form of probability, we have to evaluate it for every token in the corpus. We decide to weigh every token with its overall abundance in the corpus. This is the most immediate choice thinking of words' probability. We comment on other ways to count tokens in the Appendix B.3.

5.1 Normalisation

Given the weight, we have to find the correct normalisation. When normalising the weights to estimate P_0, we must balance two factors. On the one side, we use a continuous probability in the process. On the other is the exchangeability of the sequence. If it is impossible to extract the same token from P_0 twice, removing the weight of already observed tokens from the normalisation term is natural. The next extracted token must be chosen only among those not seen yet. This approach, however, constantly changes the P_0 and, most of all, makes it dependent on the order of the extractions.

Consider a P_0 including only the following terms and related weights: (`"the"`, 1000), (`"golden"`, 10), (`"megalodon"`, 1). The sentence:

`the golden megalodon`

will have probability:

$$\frac{(\theta \mid \alpha)_3}{(\theta)_3} \frac{1000}{1011} \frac{10}{11} \frac{1}{1} \approx \frac{(\theta \mid \alpha)_3}{(\theta)_3} \cdot 0.899$$

The sentence:

[1] A similar problem as in section 6.2.

`megalodon golden the`

instead, will have probability:

$$\frac{(\theta \mid \alpha)_3}{(\theta)_3}\frac{1}{1011}\frac{10}{1010}\frac{1000}{1000} \approx \frac{(\theta \mid \alpha)_3}{(\theta)_3} \cdot 9.79 \times 10^{-6}$$

The alternative that maintains exchangeability would be to keep the normalisation constant and assume that "*by chance*" we never get the same token twice. As this solution may seem odd, we considered a third one that reconciles the goat and the cabbage. We only change the normalisation constant between the author's process and the text. This way, we preserve exchangeability for both the author and the anonymous text separately and keep the normalisation value closer to its value by continuously updating it. The numerical results are close to the continuous update.

We will present comparisons of these three options. To better understand the differences, we will look at the probability and its effects on attribution. To evidentiate the differences between the different methods, we considered a single author and a single book for each corpus. We will look at the P_0 values of the tokens in the book that are missing in the author corpus. These are the only values from P_0 we will use during attribution. Common tokens that have already appeared will always reinforce previous occurrences due to our continuous approximation.

We compared the longest text in each corpus with the author that has the largest number of tokens. The choice of the author guarantees many different tokens. This choice implies a significant difference between the case with fixed normalisation and the other two options. The use of the longest text gives us many tokens that are absent in the reference author. This choice amplifies the difference between the continuous update of the normalisation constant and its single update, only removing the multiplicities of words appearing in the reference author.

We consider as an example the literary English corpus, and we compare *Magnum Bonum*, written in 1879 by Charlotte Mary Yonge (1823-1901), with the corpus of George Alfred Henty using *Dictionary words*. In figure 5.1 we present the value of the effective P_0^{eff} for the tokens in *Magnum Bonum* that were never used by Henty. We first notice a factor 20 parting the P_0^{eff} estimated through fixed normalisation and the other two methods. We then notice that the difference between the continuous and the single update is not constant, as expected, but neither shows a clear trend.

The continuous update normalisation introduces a bias in favour of words introduced later, see fig 5.2. As shown above, the difference can be quite dramatic depending on the order of the tokens. The maximum relative increase observed across corpora and tokenisation techniques is usually around 10% and only with the

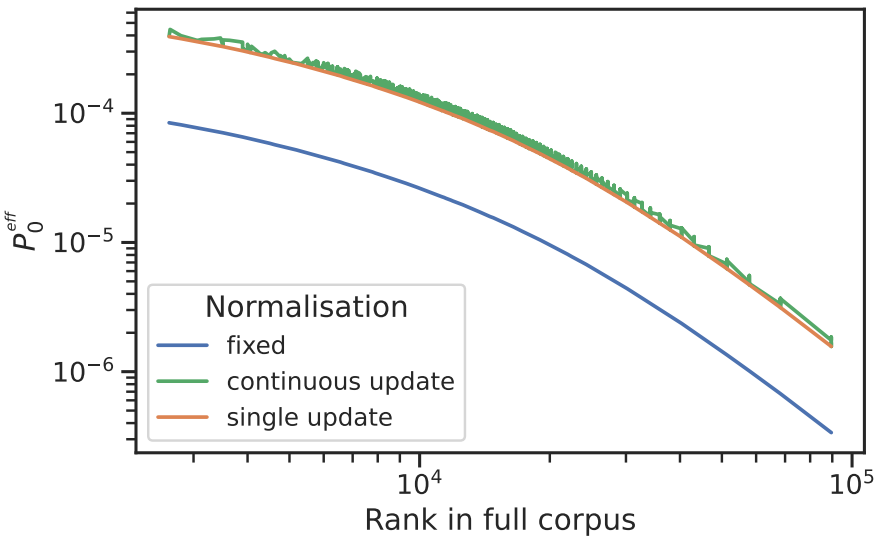


Figure 5.1. Example of effective P_0^{eff} with different normalisation procedures. The factor dividing the fixed normalisation from the single update is 19.81. No word in the 2000 most common is observed in *Magnum Bonum* but absent in Henty's corpus.

Email corpus greater than 30%. These are figures taken from the longest texts in the corpus. With shorter ones, the effect will be even smaller.

Since we are working in logspace is useful to look at the differences in

$$\sum_{i=k'+1}^{k'+k} \log_{10} P_0(y_i)$$

This difference is maximum when the tokens appear rank-ordered in the text. In this case, we would have differences usually within 2% and never above 5%. However, the tokens are introduced in an order only slightly influenced by the rank. Spearman's correlation coefficient shows a positive correlation that, even if it has minimal p-values, is extremely weak: usually smaller than 0.1, tops at 0.232 with the prolific authors of the Blog corpus. As a result, the difference in the sum of logarithms is, in most cases, smaller than 1%. This difference is often as little as the contribution of ten of the least probable tokens, comparable with the number of typos one may expect to find[2].

We can thus consider the single and continuous update normalisation as almost equivalent. Thus, we will mainly use the single update as more straightforward and exchangeable. We will focus on the difference between this and the fixed normalisation.

[2]In the hypothesis that a misspelt word will introduce a new token with multiplicity one.

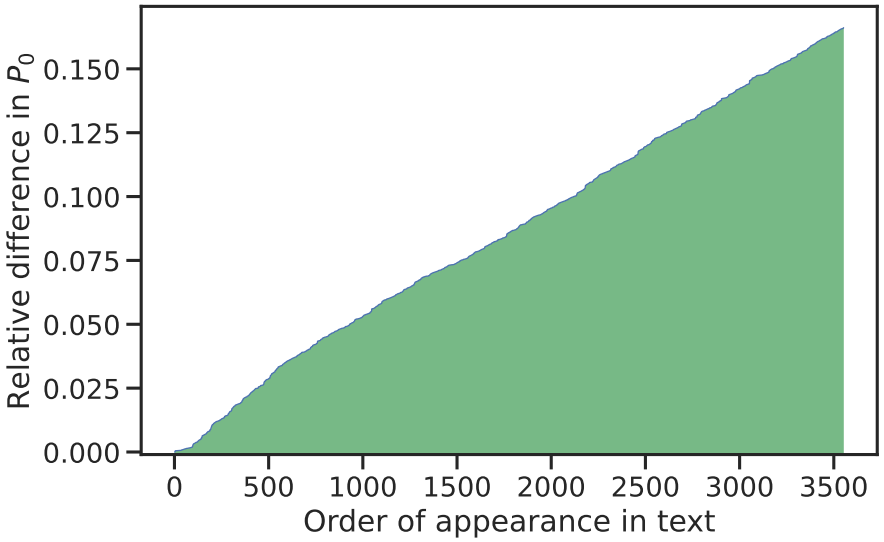


Figure 5.2. The relative difference between single and continuous update normalisation. Using continuous update normalisation, the later a token appears, the smaller is the normalisation term. The relative difference with the single update increases monotonically.

Let us now consider the effects on attribution. In figures 5.3 and 5.4 I present the best results for the English and Italian literary corpora. Results for the literary Polish and Email corpora are displayed in Appendix C, figures C.9 and C.10.

When comparing choices for the normalisation, there is no consensus on the best option. As anticipated, the single and continuous update of the normalisation give identical results when using short fragments. Only with fragments lengths $F \gtrsim 2\times10^4$, already the length of a short novel, the two normalisations are noticeably different. For all corpora, the continuous update gives worse results.

We decided to keep only the fixed and single update normalisations. We keep the choice of the normalisation as a hyperparameter. We discard the continuous normalisation for several reasons. First, from a theoretical point of view, it destroys the exchangeability of the sequence we are interested in. Second, there is little difference in the results for practical sizes of the fragments and, when there is a difference, the continuous update is worse. Third, the continuous update is also expensive from a computational point of view, requiring updating the denominator hundreds of times for long sequences.

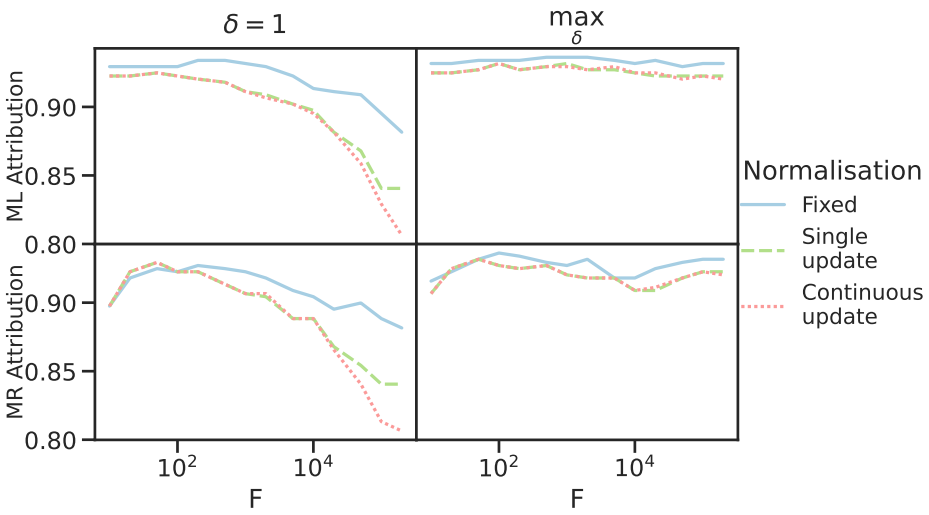


Figure 5.3. Text attribution in the literary English corpus with the different choices of P_0. We report the best scores using the different choices of P_0 in table C.4.

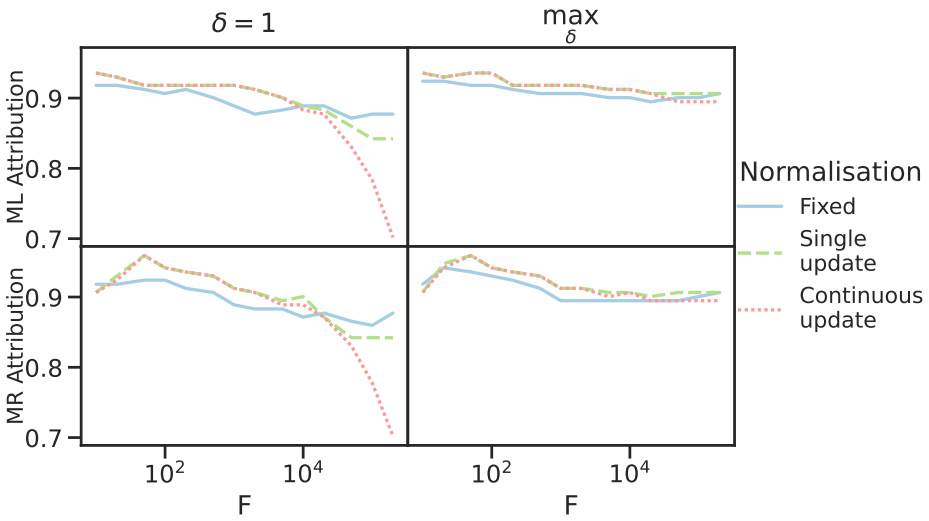


Figure 5.4. Text attribution in the literary Italian corpus with the different choices of P_0. We report the best scores using the different choices of P_0 in table C.4.

5.2 Compensating the Unknown

Despite the normalisation choice, the methods proposed to estimate the P_0 may overestimate or underestimate the probabilities we are most interested in. This is due to a combination of the finite size of the corpus and some intrinsic characteristics the task.

To understand why the specific circumstances of the task may have an impact, we have to consider which are the elements of P_0 we are interested in. As we use only conditional probabilities, only new tokens participate in the probability. The tokens with high P_0 occur often, and their probability is less affected by discrete counts. If the corpus is not intrinsically biased, the relative error in the probability estimate is typically small.

These well-estimated probabilities do not participate in the conditional probability of texts. Most of the high probability tokens are present in every author corpus and are never new in the text, so the strongest contribution from the P_0 will come from the less frequent tokens, which will be more affected by discrete effects. A couple of counts more or less makes a noticeable difference for a thirty-count token.

The error we commit estimating the probability of the many mid- and low-frequency tokens is not necessarily zero on average. We might be systematically overestimating the probabilities of the tokens. One of the effects of the finite size corpus is that we do not know how many other tokens are possible. Using a larger corpus, we may discover that the tokens observed until now are but a drop in an ocean of rare tokens. This is not unlikely due to the fat-tailed distribution of token probabilities.

However, we might be underestimating the interesting probabilities. A relatively small bias favouring some of the very frequent tokens may reduce the probability of the whole class of mid- and low-frequency ones. The corpus or the selected features may cause this underestimate.

Finding a way to compensate *a priori* for this effect is, however, problematic. This would require extrapolation of word frequencies over a hypothetical larger corpus while considering the genre, the tokenisation technique and the chosen definition of P_0. To avoid this, we chose to correct the P_0 *a posteriori*.

For this reason, in Eq. (3.2), we introduced a new parameter δ that multiplies the value of P_0. A value of $\delta > 1$ increases the probabilities and implies a reduced probability for the very common tokens whose probabilities we never use. A value of $\delta < 1$ reduces the probabilities, including the effect of tokens we do not observe.

The hyperparameter δ corrects our estimate of P_0, and we evaluate it *a posteriori* by choosing the value that maximises the attribution on the training corpus. This

same factor may correct some biases introduced by the corpus under exam or the choices made on the fragmentation of texts discussed in the next chapter.

Please note that the value of δ and the normalisation choice in the previous section are not interchangeable. For example, choosing a single update normalisation is fundamentally different from choosing a large δ value. This is because the normalisation acts on the author level. With a single (or continuous) update normalisation, the probability conditioned to different authors will depend on different P_0 normalisations. The δ parameter compensates for a global over or underestimation of the tokens weights.

In figure 5.5, we show the effect of the δ on attribution for the literary English corpus. We notice that the curve's steepness grows as the number of different tokens in the corpus. This is because, with more different tokens, δ has more opportunities to affect the probability.

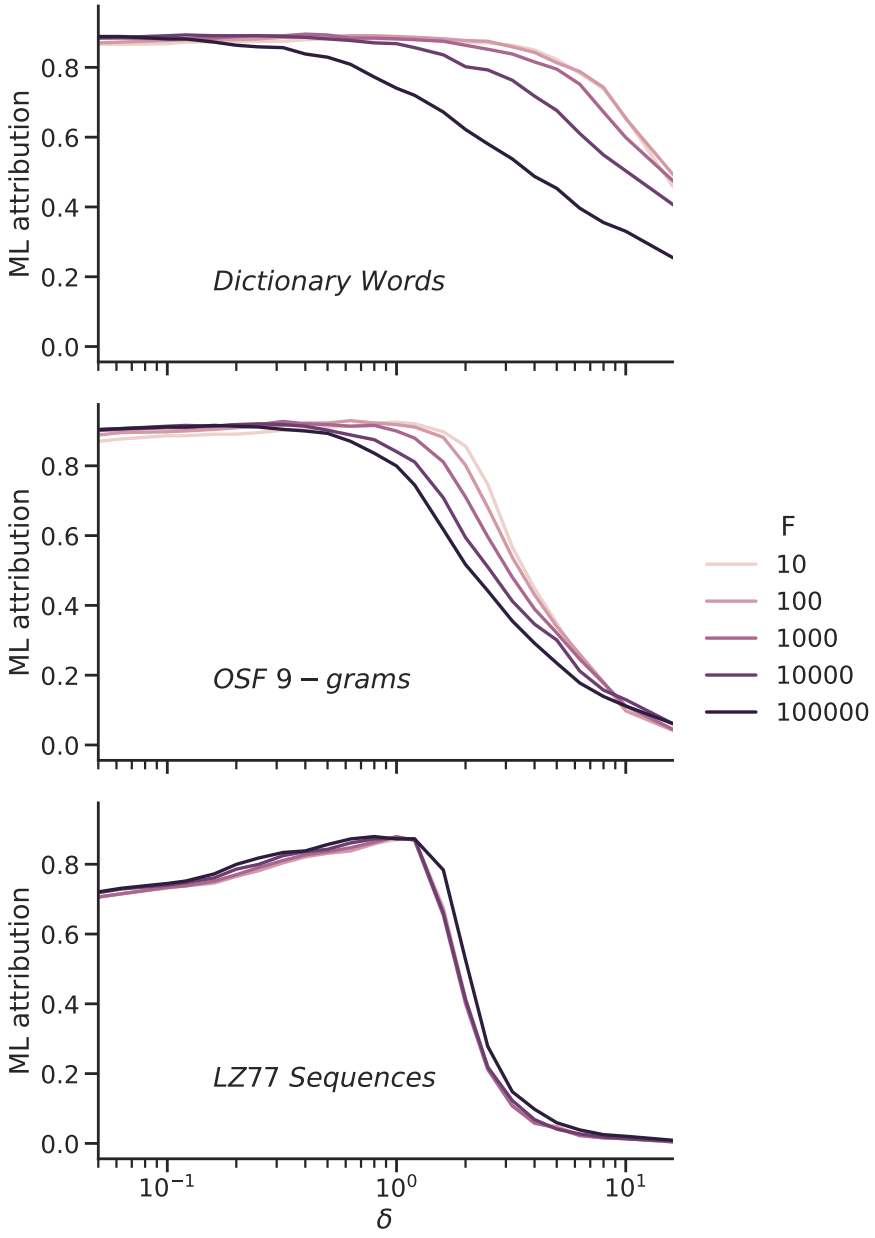


Figure 5.5. Text attribution in the literary English corpus with the different choices of variables varying δ.

Chapter 6

Choosing the Fragments' Size

We already introduced the opportunity of dividing the unknown text into fragments. This is a practical need. The token's occurrences in the text may dominate over the frequencies in the author. This happens when the text is long comparing to the author's corpus. The risk is we end up comparing the output of the unknown author with themselves.

Instead of using whole texts, we can split them into fragments. The choice to split the texts and the length of the fragments play an essential role. With very short fragments, we limit the possibility for the PD process to model the balance between retracing the past and the emergence of the new. Longer fragments exploit the specificities of the PD process at a deeper level. On the other hand, fragments too long feature the risks of long texts again.

Let us now discuss the effect of using short or long fragments. To keep the math simple, we will consider only the ML attribution in the following sections. This allows writing the probability of the sequence as the product of fragments probabilities without fragment-level assumptions.

6.1 Short Fragments

We start considering the limit of short fragments containing only tokens that have multiplicity one. To get an idea of the validity of this approximation, if we take fragments of 100 tokens, in the literary English corpus, about 60% of the tokens has multiplicity 1 when using *Dictionary words* and about 89% when using *OSF 9-grams*. In the case of *LZ77 sequences*, even if the LZ77 algorithm looks for repeated sequences, using long windows, most of the repetitions will fall in different fragments. For fragments of 100 tokens and 5.6×10^4 bytes long windows, 98% of the tokens has multiplicity 1. Predictions in this limit will continue to be guidance also for relatively longer fragments.

Let us start with the simple case where every fragment contains a single token. Using the notation from equations (3.1) and (3.2) and dividing the text f in n fragments f_i we have that:

$$P(f_i, f_i^* = y_j|\mathcal{A}, P_0) = \begin{cases} \frac{n'_j - \alpha_A}{\theta_A + n'}, & y_j \in A \\ \frac{\theta_A + \alpha_A k'}{\theta_A + n'} P_0(y_j), & y_j \notin A \end{cases} \tag{6.1}$$

for every fragment, i.e. token. Indeed, since every fragment contains a single token, it does not alter the values k' and n'_j. No token gets reinforced by subsequent recurrences in the fragment. Every token absent in A has the probability associated with its extraction from the base distribution, even if it is ubiquitous in the unknown text. The author's process cannot adapt to fragments so short. Equation (6.1) is true for every $f_i \in f$, no matter what is the value of n_j with $i \mid f_i^* = y_j$.

We now multiply the probabilities for all the fragments and group together all the elements $f_i \mid f_i^* = y_j$. Taking the logarithm we obtain:

$$\begin{aligned} \log_2 P(f|\mathcal{A}, P_0) = \sum_{j=1}^{k'} n_j \times \left[\log_2(n'_j - \alpha_A) - \log_2(\theta_A + n')\right] + \\ + \sum_{j=k'+1}^{k'+k} n_j \times \left[\log_2\left(\left(\theta_a + \alpha_A k'\right) P_0(y_j)\right) - \log_2(\theta_A + n')\right] \end{aligned} \tag{6.2}$$

Where the first sum contains the contribution from the tokens already present in A and the second from those missing, we can safely extend the first sum over all the values of $j \in [1, k']$ as n_j is null for tokens in A that do not appear in f. This expression simplifies as:

$$\begin{aligned} \log_2 P(f|\mathcal{A}, P_0) = \sum_{j=1}^{k'} n_j \log_2(n'_j - \alpha_A) + \sum_{j=k'+1}^{k'+k} n_j \log_2\left(\theta_a + \alpha_A k'\right) P_0(y_j) + \\ - n \log_2(\theta_A + n') \end{aligned} \tag{6.3}$$

We now divide and multiply the argument of the logarithms in the sums by n', this allows us to take out a term $n \log_2 n'$. Dividing everything by n and calling $\nu_j = \frac{n_j}{n}$ and $\nu'_j = \frac{n'_j}{n'}$ we get:

$$\begin{aligned} \frac{\log_2 P(f|\mathcal{A}, P_0)}{n} = \sum_{j=1}^{k'} \nu_j \log_2(\nu'_j - \frac{\alpha_A}{n'}) + \sum_{j=k'+1}^{k'+k} \nu_j \log_2 \frac{(\theta_A + \alpha_A k') P_0(y_j)}{n'} + \\ + \log_2 \frac{n'}{\theta_A + n'} \end{aligned} \tag{6.4}$$

The log probability per token takes the form of minus a crossentropy $\frac{\log_2 P(f|\mathcal{A}, P_0)}{n} = -H(f, \mathcal{A})$ between the token distributions of the author and the text. Indeed we

have that, increasing the length of the reference author, the last term tends to vanish, and the first one tends to (minus) the usual crossentropy term $\sum_j \nu_j \log_2 \nu'_j$ (see Appendix A). The middle term is a penalty term that boosts the crossentropy when the process encounters novel tokens. In the limit of an infinite reference sequence, we expect the middle term to vanish as all tokens already appeared[1] and $k \to 0$.

For any finite reference author, we are still approximating the crossentropy with a smoothing of the frequencies in the reference sequence plus a term accounting for the finite size for any finite reference author. In other terms:

$$-H(f,\mathcal{A}) = \frac{\log_2 P(f|\mathcal{A},P_0)}{n} = \sum_{j=1}^{k'+k} \nu_j \log_2 \tilde{\nu}'_j + G(\mathcal{A}) \tag{6.5}$$

Were

$$G(\mathcal{A}) = \log_2 \frac{n'}{\theta_A + n'} \tag{6.6}$$

and $\tilde{\nu}'_j$ is the smoothed frequency that depends only on $\mathcal{A}$:

$$\tilde{\nu}'_j = \begin{cases} \nu'_j - \frac{\alpha}{n'}, & y_j \in A \\ \frac{(\theta_A + \alpha_A k')P_0(y_j)}{n'}, & y_j \notin A \end{cases} \tag{6.7}$$

This result is not far from some known approaches in authorship attribution. We get the same attribution result of Eq. (6.5) by subtracting from $H(f,\mathcal{A})$ a constant term $H(f) = -\sum_{j=1}^{k'+k} \nu_j \log_2 \nu_j$ that depends only on the unknown text. We are now measuring an object with the form of a Kullback-Leibler divergence plus a correction:

$$D_{KL}(f \parallel \mathcal{A}) = \sum_{j=1}^{k'+k} \nu_j \log_2 \frac{\nu_j}{\tilde{\nu}'_j} - G(\mathcal{A}) \tag{6.8}$$

Approaches using KL divergence for authorship attribution have been around for a while. Every approach proposes a suitable smoothing to give non-zero values of $\tilde{\nu}'_j$ to tokens absent in the reference corpus. For example, Zhao and Zobel in [161] used a simpler Dirichlet smoothing[2]. We report in table 6.1 the attribution scores using single-token fragments. To our knowledge, this is the first time that a Poisson-Dirichlet smoothing has been used for this purpose.

[1] We assume that Heaps' law will continue to hold: when the length of the sequence goes to infinity, so do the number of different observed tokens and the interval between new tokens. The probability of finding a new token in the finite length text f goes to zero. This also requires the base distribution P_0 to be discrete. See section 5 for more considerations about the base distribution.

[2] The paper is focused on *Authorship Search* and the authors refer to the candidate author as a 'query'.

Table 6.1. Attribution scores using single-token fragments. DICT stands for *Dictionary words*, OSFNG for *Overlapping Space Free N-grams*, and LZ77 for *LZ77 sequences*. The best score for each corpus is highlighted in boldface.

Corpus		Score		
		DICT	OSFNG	LZ77
Literary	Polish	0.727	0.869	**0.939**
	Italian	0.702	**0.936**	0.754
	English	0.879	**0.929**	0.874
Email		0.494	**0.510**	–

Fragments with more than one token. If the fragments considered have more than one token, but every token has a multiplicity one, and no more than one token is missing in A, this result still holds with minor adjustments. Indeed if we look at Equation (6.1) we notice that with more than one token, the only difference is in the denominator, which is $\theta_A + n' + 1$ for the second token, $\theta_A + n' + 2$ for the third and so on.

We can change the definition of G in a way that depends on f only through the size of the fragments and not their content. We call s_j the number of fragments with *at least* j tokens so that $\sum_j s_j = n$ and write:

$$G(\mathcal{A}, \mathbf{s}) = \frac{1}{n} \sum_{j \geq 1} s_j \log_2 \frac{n'}{\theta_A + n' + j - 1} \tag{6.9}$$

When all the fragments contain one token (i.e. when $s_1 = n$ and $s_j = 0 \,\forall\, j > 1$) we recover the form of G in Equation (6.6). Note that we are not including any knowledge about f since the s_j depend only on how we manipulate the fragment and may be decided arbitrarily.

If more than one of the tokens in the fragment is missing in A, things get a bit more complicated. The second missing token has, for the numerator in the second case of Equation (6.1), $\theta_A + \alpha_A(k' + 1)P_0(y_j)$ that is a small alteration of $\tilde{\nu}'_j$. In many cases, this effect is small (e.g. when α_A is small, when $\theta_A \gg \alpha_A k'$, or when n' is large) and is convenient to show how Equation (6.5) changes.

We can still change G to incorporate a term depending on the number of fragments containing a certain amount of tokens missing in A. We call t_j the number of fragments with *at least* j tokens missing in A, where $\sum_j t_j = \sum_{j>k'} n_j$. We do not need to know the identity of the tokens, and it is thus possible to estimate their values to get a first-order correction without including knowledge from f. We can

rewrite the term $G(\mathcal{A}, s, t)$ to include the terms $\theta_A + \alpha_A(k' + j - 1)$:

$$G(\mathcal{A}, s, t) = \frac{1}{n} \sum_j s_j \log_2 \frac{n'}{\theta_A + n' + j - 1} + \\ + \frac{1}{n} \sum_{j>1} t_j \log_2 \frac{\theta_A + \alpha_A(k' + j - 1)}{\theta_A + \alpha_A k'} \quad (6.10)$$

If every fragment has at most one token that is missing in A, then $t_j = 0 \, \forall \, j > 1$ and the second term vanishes.

Allowing the tokens to have multiplicities other than one would require major changes to Eq. (6.7). We must introduce a correction $\mathcal{O}(\frac{n_j}{n'})$ to both cases. This correction may be relatively small in the first case of Eq. (6.7) but is usually quite big in the second. $P_0(y_j)$ is typically small for tokens that are so rare to have zero occurrences in the whole production of the reference author.

Even if the smoothing is not constant anymore, the numerical results can remain close to those obtained in our first version of Eq. (6.5). When there are few tokens $y_j \notin A$ or a few of them have multiplicity higher than 1 in f. We still have an object related to a crossentropy.

We also notice that using fragments with more than one token corresponds to a step forward beyond the assumption of independence when computing the joint probability. From this point of view, the choice of the fragments' size is an assumption on the sequence's correlation length.

In figure 6.1 we present the general behaviour of all three literary corpora and the Email corpus when dealing with short fragments, in the case of *Dictionary words*. Results for the *OSF N-grams* and *LZ77 sequences* are displayed in Appendix C, figures C.12 and C.13.

While using Maximum Likelihood attribution, the results are largely independent of the fragment size. The Majority Rule method performs poorly with short fragments. The tuning of δ cannot fix the poor results with MR. For the literary corpora, MR's results improve quickly and reach values close to the maximum already for $F \approx 10$. When the fragments are extremely short, the authors having the highest frequency for high-frequency tokens will get many fragments. Even if they give very little probability to all others, this will already guarantee the relative majority of the fragments and the attribution of the text.

In the case of the Email corpus, the improvement is slower, and the attribution reaches its highest values only when more than half of the texts are shorter than one fragment. This is due to the short author effect introduced in section 6.2, see also fig. 4.30 and its caption.

To better visualise the limits of validity of the short fragment approximation, we report in figure 6.2 the average of the conditional probability per token varying the

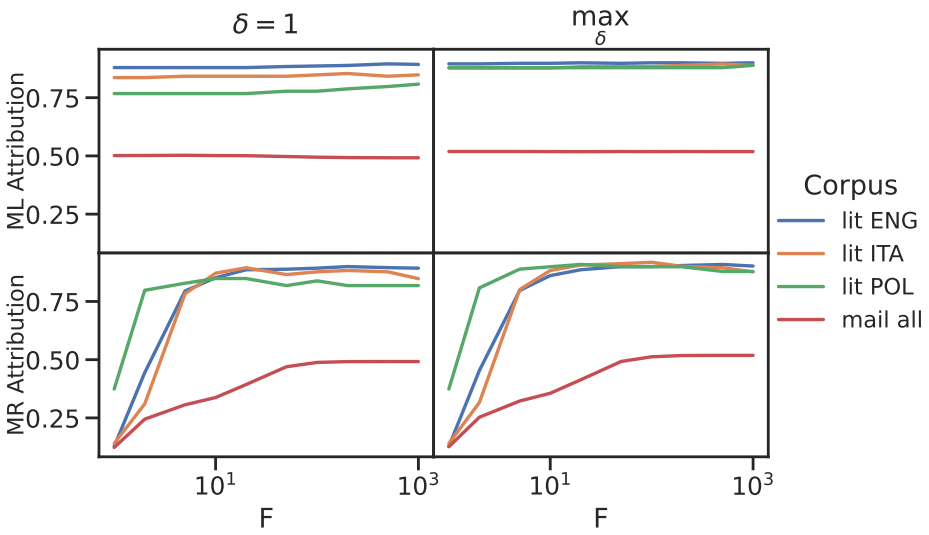


Figure 6.1. Text attribution using *Dictionary words* and short fragments. We excluded from this analysis the Blog corpora due to their size. Fragment lengths in the range $[1, 1000]$ spaced one third of decade.

fragment size over the conditional probability with fragments of size one. Results using *Dictionary words* and *LZ77 sequences* are in figures C.14 and C.15. In the latter case, the short fragment approximation holds exactly. As expected, the process "learns" from the fragment, and the conditional probability grows for growing fragment size. To lighten the notation, we will call $P(f^F \mid \mathcal{A})$ the probability $P(f \mid \mathcal{A})$ when estimated using fragments of size F.

The $\frac{1}{n} \log_2 P(f^1 \mid A)$ is of order ten for all corpora using single token fragments. The probability conditional to the best non-correct author (dashed lines) grows more than the true one. If the trend also continues for larger values of F, this can affect the attribution. However $\frac{1}{n} \log_2 \frac{P(f^1|A^*)}{\max_{A \neq A^*} P(f^1|A)}$ is usually in the range of 10^{-1}, larger than the differences observed here.

The decrease in probability for $F = 2$ is not surprising. Less than one fragment every 200 contains two equal tokens. We are still in the limits considered above where we can correct the G factor to take into account longer fragments. In the expression for the corrected factor G in eq. (6.9) we have $s_1 = s_2 = \frac{n}{2}$ and the factor becomes:

$$G = \frac{1}{2}\left(\log \frac{n'}{\theta + n'} + \log \frac{n'}{\theta + n' + 1}\right) \overset{\theta+n' \gg 1}{\to} \log \frac{n'}{\theta + n'} - \frac{1}{2(\theta + n')} \tag{6.11}$$

That is the factor G for single token fragments minus a term depending on the reference author. As the author lengths and the θs are roughly one order of magnitude smaller for the Email corpus compared to the literary corpora, this negative correction is about one order bigger.

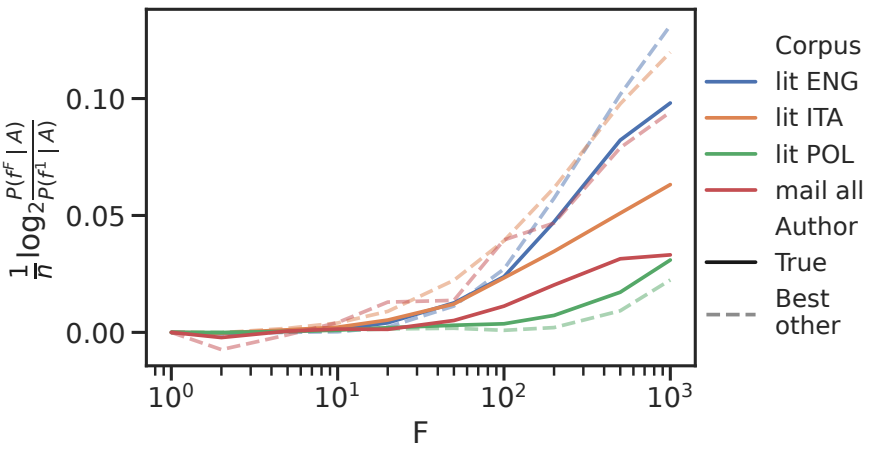


Figure 6.2. Ratio of the average conditional probability per token varying fragment size over average probability using single token fragments. Results using *OSF N-grams*. Due to outliers, the average includes only the central 95% of the values. Continuous lines show the probabilities conditioned to the actual author $\frac{1}{n}\log_2\frac{P(f^F|A^*)}{P(f^1|A^*)}$. The shaded dashed lines show the probability of the most probable of the other authors $\max_{A\neq A^*}\frac{1}{n}\log_2\frac{P(f^F|A)}{P(f^1|A)}$.

For longer fragments[3], we have – if all the fragments contain the same number of new tokens – that the assumption of having all tokens with multiplicity one would give a probability evolving as:

$$\frac{\log_2 P(f^F \mid \mathcal{A})}{n} \sim \frac{\log_2 P(f^1 \mid \mathcal{A})}{n} - \frac{F-1}{2(\theta+n')} + \frac{\alpha k(k-1)}{2F(\theta+\alpha k')} \tag{6.12}$$

This form decreases with F unless f contains many tokens missing in A[4]. The last term will favour authors with many missing tokens. Authors different from the actual one will get only the penalty for the missing tokens for short fragments. This term is the root of the faster growth of the best wrong author in figure 6.2.

For all corpora and feature extraction approaches, the relative difference between $\frac{1}{n}\log_2(f^F \mid \mathcal{A})$ and $\frac{1}{n}\log_2(f^1 \mid \mathcal{A})$ remain smaller than 1% at least up to fragment lengths in the order of the few hundreds. See also figures C.14 and C.15. The best wrong author in the Email corpus using *Dictionary words* is the only exception.

To explain the rising curves in figure 6.2, we must drop our approximation and consider the presence of repeated tokens.

[3]This approximation holds when we can approximate $\log(\theta+n'+F-1) \sim \log(\theta+n') + \frac{F-1}{\theta+n'}$.
[4]For $n' \gtrsim 4k' + (\frac{4}{\alpha}-1)\theta$, there is no possible value of $k \leq F$ such that Eq. (6.12) is increasing.

6.2 Long Fragments

A second necessary limit is when the length of the fragment ceases to be negligible relative to the reference author's sequence. In this case, the process of the reference author learns from the fragment and adapts, giving a high probability for the sequence almost regardless of the sequence of the author.

We will focus on the probability for the $m+1$-th token in the fragment. This allows understanding the effect of the growing size better than looking at the conditional probability of the whole sequence at once. The conditional probability will then include terms for small and large m with different biases towards the shorter author. The author with the maximum likelihood depends on the balance of these terms.

To show this behaviour we consider Equation (6.1) after m tokens from fragment f_i of length F have been evaluated.

$$P(f_{j,m+1}, f^*_{j,m+1} = y_j | \mathcal{A} \cup \{f_i\}_1^m) = \begin{cases} \frac{n'_j + n_j(m) - \alpha_A}{\theta_A + n' + m}, & y_j \in A \cup \{f_i\}_1^m \\ \frac{\theta_A + \alpha_A(k' + k(m))}{\theta_A + n' + m} P_0(y_j), & y_j \notin A \cup \{f_i\}_1^m \end{cases} \tag{6.13}$$

Here $n_j(m)$ represents the number of tokens corresponding to y_j in the first m and $k(m)$ the number of different tokens missing in A.

The second case refers only to the tokens missing in A and the first m tokens of f_i. This is the mark of a first physiological form of learning: when reading a new text, we are surprised the first time we find a new word but not the following.

To understand what happens with long fragments, we shall adopt a classification for the incoming tokens. Any new token must belong to one of the following three categories:

(i) common tokens – tokens with mid-high probability widespread in the language;
(ii) author tokens – rare tokens, used by the author but unlikely to appear in other authors' corpora;
(iii) document tokens – rare tokens appearing in the document that any author unlikely uses, including the actual one.

The category of a token will influence its effect when appearing in long fragments.

Let us now consider we are comparing the fragment with the actual author $\mathcal{A}^*$ and another $\mathcal{A}$ with a much shorter corpus. Every new token will fall in one of the following cases:

1. the token is new to both authors;
2. the token is known only to the short author;
3. the token is known only to the actual author;

4. both authors know the token.

Tokens from category (i) will fall mainly in the fourth case. However, some may fall in cases two or three and only a few in the first.

Tokens from category (ii) will fall in cases 3 or 4, the latter mainly if already appeared in the first m tokens of the sequence.

Tokens from category (iii) will fall mainly in the first case or case 4 if they already appeared in the first m. We can expect the larger actual author process to have $k'^* > k'$. Therefore, elements from this category are more likely to fall in case 3 than 2. After scanning m tokens in the sequence, we expect many document tokens to be known and fall in case 4.

Of the tokens appearing for the first time, we cannot say how many will be author tokens and how many document tokens. This balance will change from text to text and is one of the factors determining the ease of attribution.

We will now show how short authors are favoured in most of the four cases when the fragment is long but shorter than the author corpus. Many of the incoming tokens will fall in the fourth case in this setting. The shorter author will suffer less for the penalties associated with the extraction from P_0. If using longer fragments, being short ceases to be an advantage.

In the following, we will get an idea of the trends at play. Every author and every text will have their specificities. Even the shortest author will not gain a fragment that uses tokens only known to its author. Even the longest fragment cannot help the actual author if the fragment uses a different style.

Let us now analyse one by one the four cases:

Case 1. The shorter author is favoured when

$$\frac{\theta_A + \alpha_A(k' + k(m))}{\theta_A + n' + m} P_0(y_j) > \frac{\theta_{A^*} + \alpha_{A^*}(k'^* + k(m))}{\theta_{A^*} + n'^* + m} P_0(y_j) \tag{6.14}$$

We note that when $n' > m$, n' is the leading term in the denominator of the left member[5]. Assuming $k \propto n'^{\alpha}$ as the leading term of the numerator, we have that the short author is favoured if:

$$\frac{A}{n'^{1-\alpha}} > \frac{B}{n'^{*\,1-\alpha}} \tag{6.15}$$

The constants A and B depend on θ, α and the proportionality between k' and n.

This relation shows a general trend where the shorter author, the bigger the advantage over the actual one. However, the factors hidden in the two constants, A and B, prevent us from knowing *a priori* the direction of the inequality for any specific $\mathcal{A}^*$.

[5]From Eq. (3.5), we have $\frac{\partial P}{\partial \theta}\big|_{\theta=n} < 0$ thus $n' > \theta$.

When m becomes larger than n', it becomes the leading term in the left denominator, roughly substituting n'. So now, as m increases, the advantage of the short author decreases.

Case 2. The left member of equation (6.14) becomes

$$\frac{n'_j + n_j(m) - \alpha_A}{\theta_a + n' + m} \tag{6.16}$$

This is likely to be one to four orders of magnitude larger, and the favour for the shorter author is sharp.

Case 3. In this case, the parts of case 2 are switched, and the actual author is favoured. However, the same mechanism of case 1 might reduce this favour.

Case 4. The last case requires some care in the estimate of the advantage. Therefore, we focus on the first case of Equation (6.13): tokens that already appeared in A or in the first m tokens of f_i. First we identify

$$n_j(m) = \nu_j m$$

considering homogeneous the spatial distribution of tokens in f with $\nu_j$the frequency of y_j in f_i. This is not a big assumption since we already consider f exchangeable which implies that the homogeneous and the non homogeneous sequences have the same probability. Second, we write

$$n'_j = \nu'_j n' = (\nu_j + \Delta\nu'_j)n'$$

focusing on the difference between the frequency of tokens in f and A.

We can now write the first case of Equation (6.13) as:

$$\frac{n'_j + n_j(m) - \alpha_A}{\theta_A + n' + m} = \frac{(\nu_j + \Delta\nu'_j)n' + \nu_j m - \alpha_A}{\theta_A + n' + m} = \frac{\nu_j(n' + m) + \Delta\nu'_j n' - \alpha_A}{\theta_A + n' + m} \tag{6.17}$$

This allows us to write the probability of $f_{i,m+1}$, up to a factor $\frac{\theta_A + n' + m}{n' + m}$ independent from j, as:

$$P(f_{i,m+1}, f^*_{i,m+1} = y_j | \mathcal{A} \cup \{f_i\}_1^m) \propto \nu_j + \frac{n'\Delta\nu'_j - \alpha_A}{n' + m} \tag{6.18}$$

The probability is thus defined as proportional to the token frequency in the fragment itself plus a correction. This correction depends on the frequency difference with the reference author, the length n' of its sequence A, and the number of fragment's tokens already observed.

In this case, we are far from the clean crossentropy case of the previous section, but the general idea behind this formula is similar. We can see that if the second term is always zero ($\Delta\nu'_j = \frac{\alpha_A}{n'} \; \forall \, i \mid y_j \in f$) or in the limit $m \to \infty$, we would be measuring exactly the entropy of f. This is not possible as there are values of j for which ν'_j is null and $\Delta\nu'_j$ must be negative, and some $y_j \notin f$ have positive ν'_j and $\Delta\nu'_j > 0$ but do not enter in the calculation. This means that

$$\sum_{j|y_j \in \{f_i\}_1^m} \Delta\nu'_j < 0$$

.

The best strategy to obtain a higher probability is to keep all the $\Delta\nu_j$ as close as possible to 0. This is because the penalty paid for a negative factor is greater than the benefit obtained from a positive one of equal absolute value[6].

However, keeping the $\Delta\nu_j$ small is not enough for long fragments and short authors. To understand why, we assume that the frequencies of the tokens used by $\mathcal{A}^*$ are usually closer to those found in the text so that

$$|\Delta\nu'^*_j| < |\Delta\nu'_j|$$

for many y_j. We cannot make any assumption on their sign. However, we assume that the fragment and the actual author produce tokens according to a common ideal frequency. Thus the differences are due to discretisation and

$$\mid \Delta\nu'^*_j \mid \sim \frac{1}{F} \; \forall j$$

For the short author we make no assumptions and rewrite the frequency difference as:

$$\mid \Delta\nu'^*_j \mid = \left| \frac{n'_j}{n'} - \frac{n_j}{F} \right| = \frac{1}{F} \left| \frac{n'_j F - n_j n'}{n'} \right| = \frac{C}{F} \tag{6.19}$$

If the winning strategy is to keep the absolute value of the fraction in Eq. (6.18) as small as possible, the shorter author will be favoured if

$$\left| \frac{n' \frac{C}{F} - \alpha_A}{n' + m} \right| < \left| \frac{n'^* \frac{1}{F} - \alpha_{A^*}}{n'^* + m} \right| \tag{6.20}$$

When $n'^* \gg n' > F > m$, we can ignore the α in the numerators and m in the denominator of the right term. The condition becomes

$$\frac{\mid C \mid}{1 + \frac{m}{n'}} < 1$$

and it is true for some $\bar{m}$ when $\mid C \mid < 2$ or for less ideal hypotheses for $\Delta\nu'^*_j$.

[6] This is straightforward considering that $\log(1-x) < -\log(1+x)$ and the bigger is x, the bigger is the difference.

The effective frequency appearing in (6.18) will be closer to ν_j for the short author $\mathcal{A}$ than for the true one $\mathcal{A}^*$. The positive and negative displacements cancel out better for the shorter author leaving the longer with a negative bias. The term $\frac{n'+m}{\theta_A+n'+m}$ we left back in Equation (6.18) is smaller than one and tends to reduce the probability more for smaller n'. However, since its dependency from θ_A, if $\theta_{A^*} > \theta_A$ there is no guarantee that this term will favour the actual author instead of the short one. In any case, this term cannot cancel this effect but only increase the threshold $\bar{m}$.

As the fragment length grows, more and more y_j may favour the short author instead of the right one up to a point where the correct author loses compared with a much shorter one. This effect is quite noticeable as more and more fragments – and then texts – are assigned to the shorter author, growing the fragment's size.

This mechanism is very effective in the case of text-specific tokens where $\Delta\nu'_j = \Delta\nu'^*_j = \nu_j$. In this case, the previous condition approximates to

$$\frac{1}{1+\frac{m}{n'_j}} < 1$$

and is always true.

The bias can also affect author tokens. Those we defined as author tokens are fairly uncommon tokens. Except for extreme cases this is required if we want only a few authors using them[7]. Consider an author token appearing twice in $\{f_i\}_1^m$. The frequency of this token in the fragment is not smaller than $\frac{2}{F}$ and may be much higher than the frequency in the author. The true author avoids the second case of equation (6.13) and the small P_0 associated with the first occurrence. Nevertheless, $\Delta\nu^*_j \approx -\nu_j$ and the second occurrence may favour the shorter author.

Things change when the frequency is small, and the fragment has size $F > n'$. Therefore, we continue to focus on the following occurrences of tokens missing in the short author corpus. For these tokens the frequency will have values $\nu_j = \frac{n_j}{F}$ with $n_j = 2, 3, \ldots$ but small. The numerator in Equation 6.18 is now dominated by α_A. Longer authors will benefit from higher m and more fine-grained ν_j. For the shortest author the effective frequency won't grow more than $\approx \nu_j - \frac{\alpha_A}{F}$. The longer the fragment, the higher the values of n_j that stop favouring the shortest author.

When fragments become long enough, the short author no longer benefits from its fast learning. At the same time, it continues to be penalised by the presence of author tokens. If the number of author-specific tokens is large enough, the actual author will be preferred again.

[7] This is true at least for *Dictionary words* and *OSF N-grams*, see figures 4.3 and 4.16. The relative amplitude of the fluctuations in token frequency between authors increases as the frequency decreases. Less common tokens are unevenly distributed across authors.

To summarise, we identify three regions. First, we have the region for small m. The missing author-tokens, the small k', and the different frequencies penalise the short author. Then, in the $1 \ll m \lesssim n'$ region the short author is favoured. Finally, a region for $m \gtrsim n'$ where the bias changes direction. The results will depend on the number of tokens from the different regions.

We will now show how the presence of short authors affects the results and how it is related to the chosen tokenisation. To amplify the effect of short authors, we chose to consider only the literary corpora. Here authors are, in general, large. None has less than 1.2×10^5 tokens with any preprocessing method.

We want to isolate the effect of the author length from the characteristics of the author itself. To obtain this, we need many short authors of the same size. We removed the five shortest authors from each corpus. We tokenised the books of these five authors, and we joined them to get five author sequences. Then, we divided the sequence of each author into parts. Each one of them will play the part of a synthetic author. The size of the synthetic authors depends on the shortest author left in the corpus. We took synthetic authors with lengths $\frac{1}{3}$, $\frac{1}{6}$ and $\frac{1}{12}$ of the shortest author left in the corpus. We extract up to 4 synthetic authors from each author sequence. We take disjoint sections of the author sequence to reduce correlation and sample the whole author. Sometimes this is not possible, especially with long synthetic authors. In this case, we took equal size disjoint sections shorter than we needed and then fill with randomly sampled tokens from the sequence. In the case of very short authors, we derived less than four synthetic authors avoiding using many sampled tokens.

We attributed each book left in the corpus, comparing it to all the authors left in the corpus plus one of the synthetic authors at the time acting as a distractor. For every fragment size, we present the average fraction of correct attributions for the three lengths of synthetic authors.

In figure 6.3 we report the attribution scores for the literary English corpus using *OSF N-grams*. Without optimisation over δ, the performance decreases when using longer fragments. When the shortest author is shorter, this effect is more substantial and visible for even smaller values of F. This behaviour is in line with the expectations.

The impact of the short author effect depends both on the corpus and on the chosen preprocessing. Using *Dictionary words*, fig. 6.4, we get differences up to 30% while using *OSF N-grams*, fig. 6.5, these are much smaller, 5% with the literary English corpus, non noticeable with the others. In the case of *LZ77 sequences*, fig. 6.6, the differences are even stronger as the effect is very slight (2%) for the literary English corpus and large with the other two. If the shortest one is short

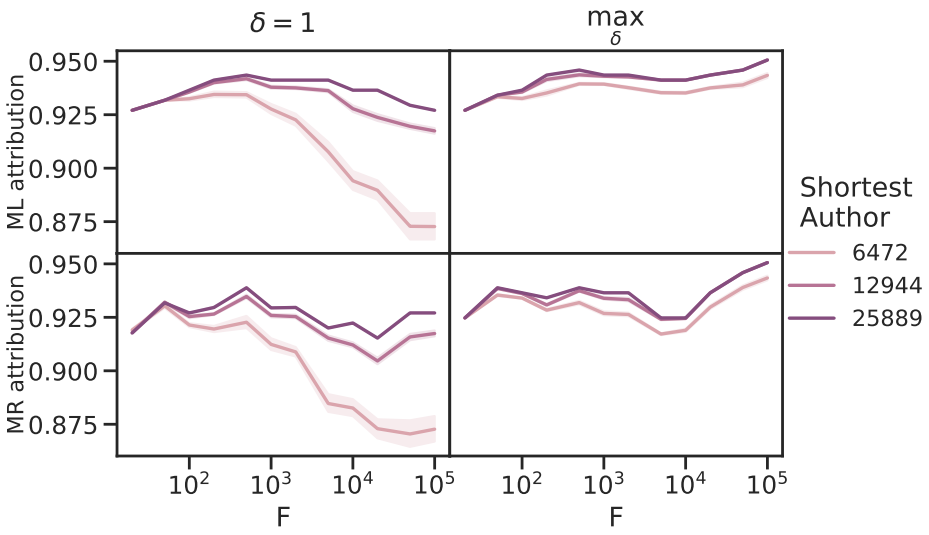


Figure 6.3. Attribution using long fragments and short authors using *OSF N-grams* for the literary English corpus. Despite the strong effect of the maximisation over δ, the losses due to the shortest author are not fully compensated.

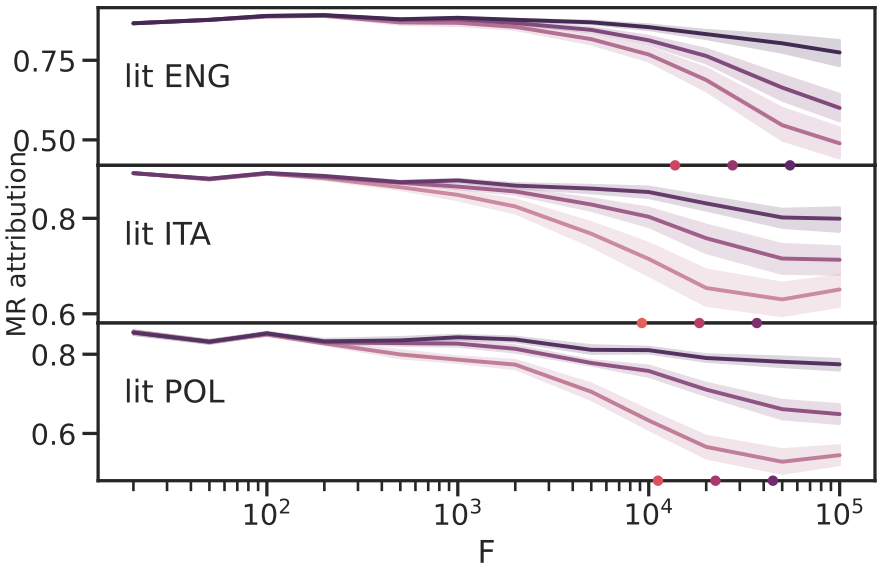


Figure 6.4. Attribution using long fragments and short authors using *Dictionary words* for the three literary corpora. The dots on the x-axis of each panel mark the length of the shortest author for the curve of the corresponding colour. Texts attributed using Majority Rule without tuning of δ.

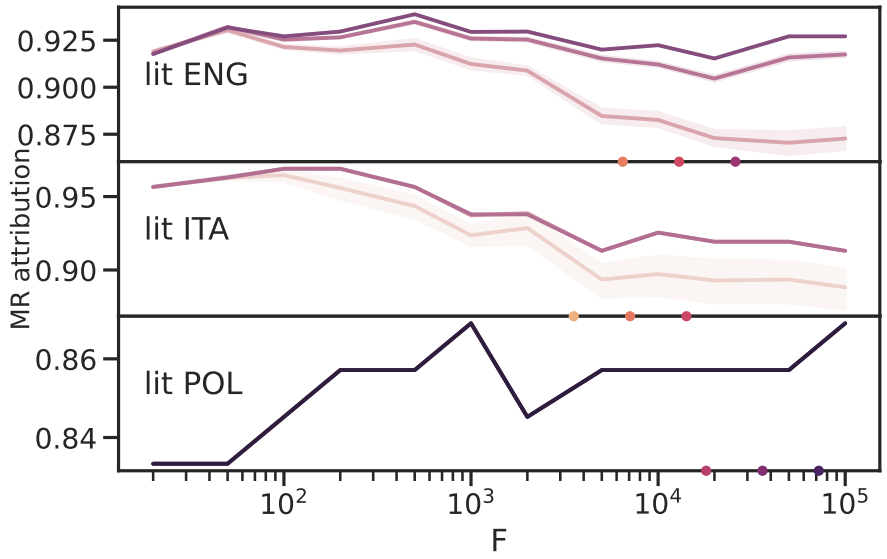


Figure 6.5. Attribution using long fragments and short authors using *OSF N-grams* for the three literary corpora. The dots on the x-axis of each panel mark the shortest author's length for the curve of the corresponding colour. Texts attributed using Majority Rule without tuning of δ.

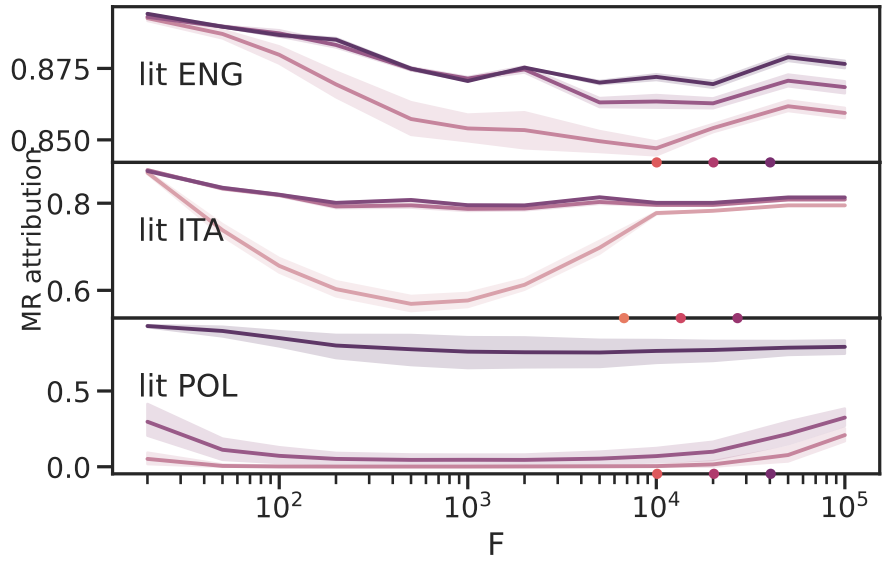


Figure 6.6. Attribution using long fragments and short authors using *LZ77 sequences* for the three literary corpora. The dots on the x-axis of each panel mark the shortest author's length for the curve of the corresponding colour. Texts attributed using Majority Rule without tuning of δ.

enough, even zero books are attributed to the right author in the Polish literary corpus.

In table 6.2, we present a summary of the difference induced by the varying length of the shortest author. We compare the results with the longest synthetic authors to those obtained with the two shorter lengths.

Table 6.2. Effect of the synthetic author length. The columns "Fragment length" report the minimum fragment length so that the results with the short or intermediate synthetic authors are significantly worse than with the long synthetic authors. The columns "Maximum difference" report the maximum over F of the difference between the average score with the long synthetic authors and the average with the shorter ones. When there is a difference in the results using synthetic authors of different lengths, the intermediate length allows agreement with the long one up to larger values of F and a (usually at least two times) smaller difference in score. Values derived from the MR attribution without tuning δ except for the last two columns.

Variable	Corpus	Fragment length		Max difference		$\min_\delta$(Max diff.)	
		Short	Interm.	Short	Interm.	Short	Interm.
DICT	English	2000	1×10^4	0.124	0.062	0.004	0.002
	Italian	1×10^4	5×10^4	0.110	0.040	0.053	0.025
	Polish	2000	1×10^4	0.242	0.084	0.094	0.055
OSFNG	English	20	100	0.099	0.023	0.014	0.003
	Italian	200	—	0.028	0.001	0.013	0.001
	Polish	—	—	0.001	0.001	0.017	0.004
LZ77	English	100	5000	0.025	0.009	0.026	0.014
	Italian	20	—	0.239	0.012	0.169	0.077
	Polish	20	20	0.751	0.709	0.375	0.284

We relate these behaviours to the properties of the extracted tokens. Using *Dictionary words* tokens are usually repeated (few *hapax* and *dis legomena*) and the effects from case 4 are strong. Using *OSF N-grams*, fewer tokens repeat, and author tokens get more weight. Low-frequency tokens – those belonging to classes (ii) and (iii) – have higher fluctuations in frequency across authors. This means that they are used quite a bit by a few authors and very little or not by the others. In this case, the effect of short authors is minimal. Using *LZ77 sequences*, due to their nature, the number of repetitions is more significant, and missing words get a smaller weight. Soon, many tokens will be already seen and fall in case 4. Therefore, these tokens will likely favour the shortest author.

We expect things to change when the length F of the fragment surpasses the length of the shortest author. As discussed above, this change will be bigger in cases

where the attribution is influenced more by low frequency (or new) tokens. In all cases where the effect of the different short author sizes is evident, the score stops decreasing or even increases when the size of the fragment is about the order of magnitude of the shortest author. As expected, this is particularly evident when using *LZ77 sequences* in all three corpora, fig. 6.6.

Another point of view on this phenomenon is the number of tokens that falls in fragment whose most likely author is the shortest one. While for some corpora the attribution procedure may mask the role of the shortest author, in figure 6.7 we can observe the clear dependence both from the fragment and the shortest author length.

The effect for the literary Italian and Polish corpora using *OSF N-grams* was small or invisible, looking only at the attributions (fig. 6.5). We see clearly how the difference in the number of tokens assigned to the shortest author follows the expected behaviour. With shorter synthetic authors, more fragments are assigned to the shortest one, even if this does not affect the attribution. The graphs for *Dictionary words* and *LZ77 sequences* are available in Appendix C, figures C.16 and C.17 respectively.

6.3 Authors' Slicing

Given the above considerations, there are different ways to mitigate the adverse effects of short authors. One is to use short fragments to ensure that, in the vast majority of the cases, they are shorter than $\bar{m}$. A second might be to use fragments so long to surpass the shortest authors. Finally, a third way to mitigate the effect of the short authors is to reduce the length differences between authors. In this way, none is much shorter than the others, the terms n' and n'^* are similar, and $\bar{m}$ is increased.

Short or long fragments have a few drawbacks that make the third option attractive. As we saw in an earlier section, with very short fragments, the process has no time to learn from the new text, and we are effectively using the PD process as a smoothing. Imagine a blogger usually writing about fishes and shrimps who post about dolphins. With short fragments, every time we find the word "dolphin", we consider it a new word with associated low probability. This continuous surprise introduces a bias favouring authors with higher α_A and θ_A. At the same time, if the shortest author is short enough, there might not be a reasonable fragment size smaller than most $\bar{m}$.

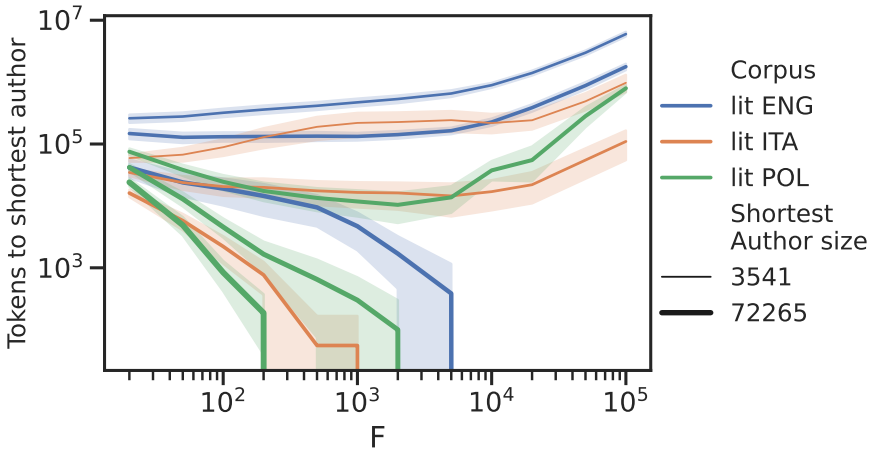


Figure 6.7. Averages for the number of tokens in fragments attributed to the shortest author using *OSF N-grams*. The weight of the lines is proportional to the length of the shortest author.

Using very long fragments may not be practical either. It may favour other authors with a small production still longer than the fragment. This is probably the mechanism at play for the informal corpora.

The results obtained in chapter 7 seem to point in this direction. We obtain the best results for the Italian literary corpus for fragments lengths in the order of the tens, up to one hundred. Given the use of 10-grams, these lengths correspond to a few sentences up to a few paragraphs. For the other literary corpora, the fragments correspond to a few pages of the original texts. In the cases of informal corpora, the best results are for fragment lengths above the length of 95% of the documents.

These results are in line with the above considerations. The Italian literary corpus and the informal corpora feature some authors much shorter than the others. In the case of the Italian corpus, it is only one author, Parrella, and the length of any of his two documents is in the order of the hundreds of thousands of characters. In this case, using short fragments may help avoid spurious attributions to Parrella. For the informal corpora, the shortest authors have a total corpus in the hundreds or the few thousands of tokens. In this case, the only effective strategy is to use fragments as long as possible. For short texts, this length may not be enough.

Slicing the authors is an interesting alternative. Dividing the most extended authors into parts may be enough to contain the difference between the n', avoiding the need to reduce the size of all the fragments.

This can also have a second positive effect. We compare each fragment, or text, with different slices of an author. If their production is vast and diversified, the text may show a good agreement with a part of it but not so much with the whole.

Obtaining more than one measure of the likelihood for the same author may allow, during attribution, to combine these values in a way that maximises the attribution.

To test this hypothesis, we selected a length of the fragments F that emphasises the differences in attribution varying the length of the shortest author. When evaluating the likelihood of each text, we sliced the reference author into chunks of different lengths. We used the length of the shortest author not excluded (three times the largest synthetic authors), two-thirds of its length, one-third (the largest synthetic authors) and one-sixth (the intermediate synthetic authors). We then combine the information from all the slices (see section 3.4 for ways to do that) to attribute each text.

Using *Dictionary words* the results are encouraging. In figure 6.8 I report the results for the literary Polish corpus. In this case, using shorter slices reduces the differences induced by the presence of short authors. With the shorter size of the slices, the most significant differences in the attribution are due to the attribution procedure itself. The more weight is assigned to the slice that gives the highest conditional probability, the better is the attribution.

In this case, the tuning of the δ parameter could not fully compensate for the differences induced by the presence of short authors. The tuning of δ has little effect (if any) when using sliced authors. The use of slices improves the results even more and reduces these differences. These results are promising to obtain even better performance without suffering from imbalances in the corpus.

Using different tokenisation procedures the picture is less clear. In figures 6.9 and 6.10 I report the results for the literary Italian corpus using *OSF N-grams* and *LZ77 sequences* respectively. When using *OSF N-grams*, we observe for all corpora a decrease in performance when using shorter slices. Not every combination of corpus and shortest author length has a slice size that improves the results (see the literary English corpus in fig. C.18). Similar considerations also work for *LZ77 sequences* where the best results are not always for shorter author slices. The slicing does not fix the extremely poor results with short comparison authors[8].

To understand the reason for these different effects, we look again at the tokens we are using and what makes the attribution effective. If many tokens are repeated, we expect improvements from the slicing of authors. Splitting the reference author should reduce the author length effect. On the other hand, the splitting increases the chances of missing a token in the reference slice. The first case of Eq. (3.2) – the one introducing δ – introduces a penalty for tokens missing in the reference author. If many tokens are missing and the difference in missing tokens between the

[8] In the case of the literary English corpus results using *LZ77 sequences* are more similar to those with *Dictionary words*, see C.19. In this case, however, the results with different synthetic author lengths were already similar.

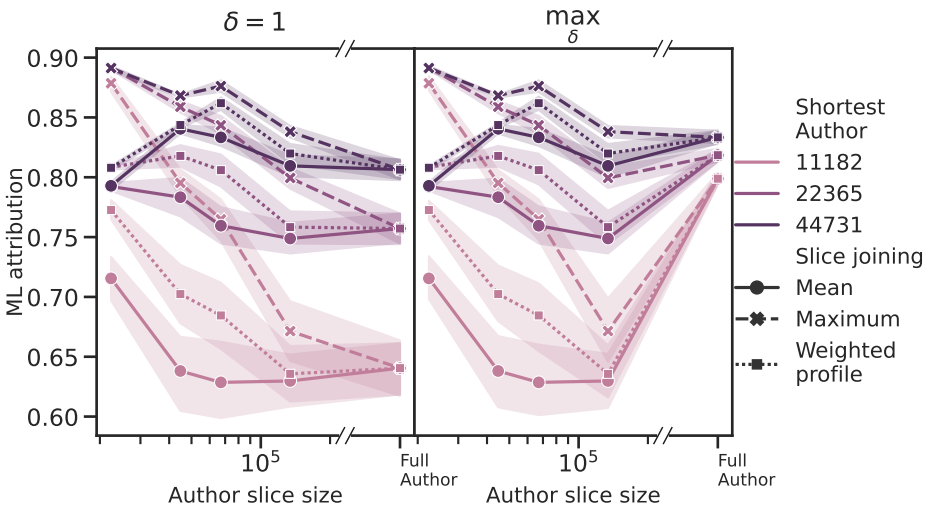


Figure 6.8. Attribution in the literary Polish corpus varying the size of the author slices using *Dictionary words*. The fragment length is fixed at $F = 10^4$. We do not report the results using MR attribution for conciseness as very close to those using Maximum Likelihood.

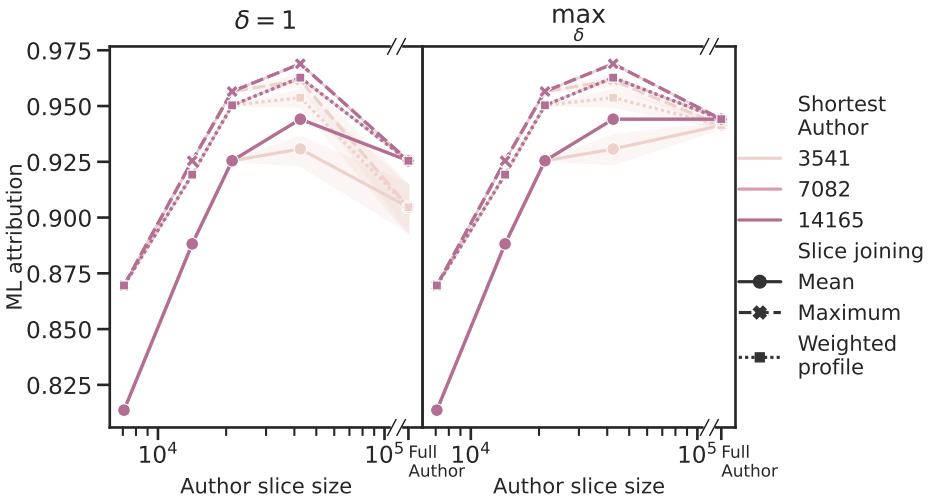


Figure 6.9. Attribution in the Italian literary corpus varying the size of the author slices using *OSF N-grams*. The fragment length is fixed at $F = 10^4$. We do not report the results using MR attribution for conciseness as very close to those using Maximum Likelihood.

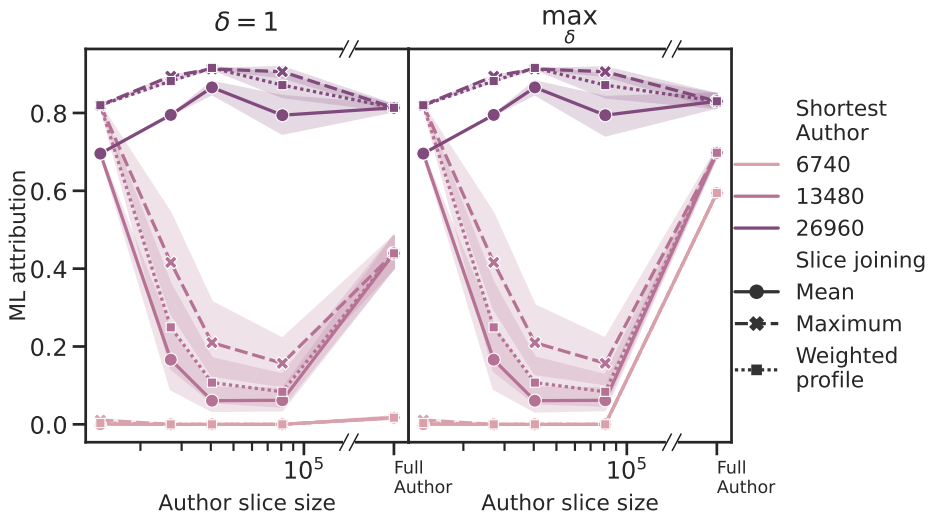


Figure 6.10. Attribution in the Italian literary corpus varying the size of the author slices using *LZ77 sequences*. The fragment length is fixed at $F = 5 \times 10^2$. We do not report the results using MR attribution for conciseness as very close to those using Maximum Likelihood.

actual author and the others is significant, we expect a strong effect from case 3 in section 6.2 and δ favouring the actual author.

In figure 6.11 I show the fraction of tokens missing in the reference authors for the Italian literary corpus under different tokenisations. In all three cases, the slicing of the reference author corpus increases the fraction of missing tokens and reduces the difference between the actual author and the others. This reduces the power to discriminate the authors.

We notice again a difference between features. Looking at the fractions of tokens missing in the reference slice with different tokenisations, we notice that there is roughly a factor five between the fractions using *Dictionary words* and the other two methods. Using *Dictionary words*, as already discussed, we observe more repeated tokens and some of the effects of the presence of short authors are more pronounced. However, these same effects can be reduced with the slicing, which leads to a more significant improvement in the results.

The Blog corpus – prolific authors is an interesting candidate for author slicing. Selecting only prolific authors, none is so short to require very small slices. However, there is a fraction 7% of texts that are one-fortieth or longer than the shortest authors in this corpus. These texts may suffer from the short author effect (see fig. 6.4) and thus benefit from the slicing. We tested this hypothesis by comparing the results using full authors or authors sliced to the size of the shortest one. We tested *Dictionary words* as this kind of feature is more affected by the short author

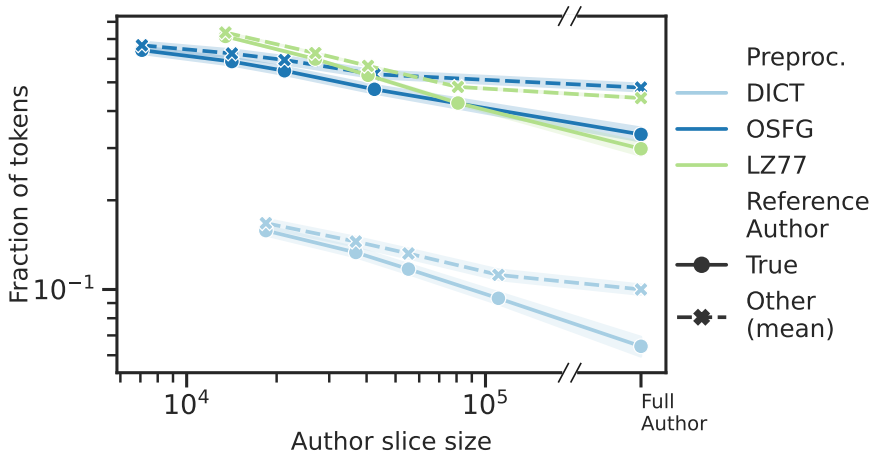


Figure 6.11. Tokens missing in the reference author slices varying their size. Results using the literary Italian corpus. Fraction of missing tokens averaged over all the slices of the author. Fragment lengths are $F = 2 \times 10^4$ using *Dictionary words*, $F = 10^4$ using *OSF N-grams* and $F = 5 \times 10^2$ for *LZ77 sequences*

effect and *OSF N-grams* as providing the best scores. The best score using *Dictionary words* is 0.475 using ML attribution and taking the likelihood of each fragment as the maximum over the different author slices. However, this result is worse than the 0.483 we obtained keeping whole authors. For the *OSF N-grams*, we get the best score of 0.483 , even in this case worse than the best score of 0.495 without slicing.

Slicing the authors is thus not a valuable alternative to choosing the correct fragments' size.

Chapter 7

Comparisons

To test our technique in a realistic setting, we will look at our corpora without taking advantage of the specific knowledge we got. We will ignore the best variables or the best normalisation of P_0 we found in chapters 4 to 6 and keep only the general guidelines we obtained. Thus, we will search for the best set of parameters in every attribution task.

As shown in sections 4.4 and 5.1, the results with different variables or choices of P_0 are pretty stable around the maxima. This stability simplifies the search. We will divide the corpora in training and test sets. For every preprocessing, we will search on a range of free parameter values and fragments lengths using the single update normalisation of P_0. Next, we search for the maximum over the δ parameter for every set of hyperparameters. We then select the best results on the training set and check if we can improve them using the fixed normalisation P_0 and optimising δ. Finally, we search the neighbourhood of the maxima for the best set of parameters, using the selected P_0 normalisation and finding the value of δ that maximises the results.

We compared the results of our technique with the results published in the literature. We selected comparable techniques and two different methods proposed in the past years. We need an active comparison with different methods, mainly for the literary corpora. We assembled the English corpus for this specific task and used the Polish corpus differently than its design goal. Only the Italian corpus was designed and used for authorship attribution. However, the original focus was on the figure of Elena Ferrante (whose books we excluded), and few comparable results are available.

The first method we used is the Crossentropy method proposed in [15] as improved in [89]. The second is, with minor edits, the Latent Dirichlet Allocation plus Hellinger distance (LDA-H) method proposed in [134]. For the sake of clarity, we will briefly describe the two methods and the eventual deviations from the original formulations.

Crossentropy The method is based on the LZ77 compression algorithm [164] described in 4.3. The authors use the compression algorithm to measure the remoteness between two texts exploiting the link between its output and the crossentropy of information sources (see Appendix A).

Given two texts A and B emitted by two different "sources" (i.e. two different authors), one can use an LZ77-like compression scheme to encode B given the best code of A. The authors in [89] achieve so scanning the B text and looking for matches only in the A text. They estimate the crossentropy $H(B, A)$ of B with respect to A. If the two texts are similar, $H(B, A)$ will be small. On the other hand, if B and A are very different, the knowledge of A will not help to encode B, and $H(B, A)$ will be correspondingly large. While $H(B, A)$ is not a distance from the mathematical point of view[1], is used as a measure of remoteness between A and B.

The authors of [89] describe an Authorship Attribution procedure based on remoteness between two texts. Take an anonymous text X and a corpus of documents of known authorship. We will denote the texts in the corpus as Y_i, where Y is the author. The index i is relative to the list of texts from Y. Instead of going directly to the measure of $H(X, Y_i)$ for all the Y_i, the authors split long texts into fragments. This division avoids having an interesting signal masked when measuring the anonymous text's average extra bits per character.

The authors compute the crossentropies $H(X_j, Y_{ik})$ of all the pairs of fragments of the anonymous text X_j and all the fragments of the texts in the corpus Y_{ik}. They then propose different ways to weigh and average the crossentropies to get the best candidate Y to be the author of the text X. The different attribution procedures used are similar to those described in section 3.4 with particular reference to the Weighted Profile and Majority Rule.

This method is a derivation of the one described in [16], and recently the authors in [108] recognised it as one of the algorithms giving the best performances out of the fourteen the authors surveyed. This method – interesting for its effectiveness and from a theoretical point of view – bears two significant shortcomings. The first is intrinsic: the LZ77 algorithm is asymptotically optimal but approaches this limit slowly. This means that the texts or fragments compared have to be quite long to achieve a reliable estimate of the crossentropy. The *gzip* algorithm uses a window about 4 kB long as it already offers a good compress ratio. This length is acceptable when working with literature where usual texts are tens or hundreds of times longer. Using windows 4 kB long, even if the entropy limit is still far, the measures are already stable, and the authors' rank is not likely to change. However, applying this method to other genres, such as poetry or informal texts, is challenging. In these

[1] In [15] the authors describe how to transform it in a distance practically.

contexts, texts are often a few lines to a few words long. The second limit of this method is in the computational complexity and execution time that scales with the total length of the corpus documents times the length of the anonymous text. In case of a leave one out attribution of the whole corpus, it may quickly reach the order of the days.

For these two reasons, we refrained to use this comparison on the Blog and Email corpora (see section 3.2) as large or made of short texts.

LDA-H The Latent Dirichlet Allocation plus Hellinger distance [134] achieved state-of-the-art results in 2011. Although it may look a bit outdated, and indeed many more complicated methods have surpassed it over the years, we chose it for comparison as its model is not too complex compared to ours.

The main idea behind the LDA-H approach is to use the Hellinger distance between document topic distributions to find the most likely author of a document. The Hellinger distance of two topic distributions is defined as:

$$D(\theta_1, \theta_2) = \sqrt{\frac{1}{2} \sum_{t=1}^{T} \left(\sqrt{\theta_{1,t}} - \sqrt{\theta_{2,t}} \right)^2} \tag{7.1}$$

where θ_i is a T-dimensional multinomial topic distribution, and $\theta_{i,t}$ is the probability of the t-th topic.

The authors propose two variants of the model:

1. Multi-document (LDAH-M). In this instance-based version, distances are computed between the topic distribution of the anonymous text and all the known texts. The author whose texts are on average closer to the anonymous one is returned as the most likely author of the test document.
2. Single-document (LDAH-S). All the documents from each author are chained together in a profile document. The proposed author for the anonymous text is the one whose profile document has the closest topic distribution to the anonymous text.

Of the two methods, we chose the second as the authors of [134] show it performs better when the number of authors is in the tens or above.

The model learns the word distribution for each topic and each document's topic distribution (i.e. the author) from the corpus through collapsed Gibbs sampling. The Dirichlet distributions, priors over the categorical distributions of words and topics, are "collapsed out". The parameters of the Dirichlet distributions are of no practical interest, and are marginalised out. The sampling reduces to that of a Dirichlet-multinomial distribution. This final form is simpler and faster in convergence than the full Gibbs sampling. The sampling takes $\mathcal{O}(WT)$ for each sample where W is

the number of tokens, and T is the number of topics. For the LDA parameters the authors followed [57] and the recommendations in the software's documentation: $\alpha = \min(0.1, 50/T)$ and $\beta = 0.01$. Here α is the concentration parameter of the T-dimensional symmetric Dirichlet distribution over the T different topics, β is the concentration parameter of the symmetric Dirichlet distribution over the words for each topic.

Regarding the number of samples, we found that taking the document representation on more than one sample gave better results (as indeed expected, see [146]). Thus, instead of [134], we followed the procedure the authors used in [133]. We run 4 chains with a burn-in of 1,000 iterations. After the burn-in, we took 8 samples spaced 100 iterations. The authors used the LDA implementation from LingPipe [5] that looks discontinued since 2011. In our implementation we adapted the `lda` module for Python [32] which in turn cites [18, 57, 115] for its implementation. The distributed version of the `lda` module does not allow for further training after the first iterations and the first sampling. We had to adapt the module to allow new training steps after the burn-in. To estimate the topic distribution of the anonymous text, instead of the procedure described in [133], we used the built-in method provided by the `lda` module that cites [22, 153].

When comparing with the other methods, we used ten-fold stratified cross-validation except for the Email corpus, see section 3.2, that provides already train, validation and test sets. In k-fold stratified cross-validation, the dataset is divided into k parts, trying to keep the share of each class (in our case, the number of documents from each author) constant across the parts. The subdivision in folds was the same with the different methods.

For a better perspective on the results, we present six different performance measures. These measures are three standard figures of merit:

- precision, P;
- recall, R;
- F1 score, F1,

in their micro and macro averages.

The precision is the fraction of an author's document in the total number of documents they get assigned. In other terms, the true positives over the total, true and false, positive for the author class.

The recall is the fraction of the author's production assigned to him, i.e. the true positives over the total author documents, true positives and false negatives.

The F1 score is the harmonic mean of the former two quantities. It is regarded as a better overall indicator as it weighs more the smallest between precision and recall.

For each of these measures, we can take the arithmetic average over all the authors (macro average) or weigh the average with the number of documents by each author (micro average). These two averages capture two different pieces of information. The macro average tells the quality of the attribution on any author (supposing no bias of the method on authors' productivity). The micro average tells the quality on any book. Comparing the two averages, we quickly estimate the bias towards more (less) prolific authors.

When computing the score – the fraction of correctly assigned documents – in previous sections, we measured the micro averaged recall. This way, we approach the problem from an "academic" point of view. We focus on questions like "is this book from Plato?" trying to avoid false negatives. On the other end, using precision would have imposed a more "judicial" point of view. In that case, the question would sound like "did John write this letter?" trying to avoid false positives.

Using a micro averaged measure as guidance maximises the total number of attributions tolerating if some less prolific authors are not well represented. No surprise then if the micro averaged figures in the results are consistently better than the macro averaged.

In figures 7.1, 7.2, and 7.3 we present a box plot of the values of precision, recall and F1 for the literary English, Italian, and Polish corpora respectively.

In table 7.1 we report the average scores for all three corpora and methods. The results using our method are, in general, better than using the other two. There is a 21.7% fraction on all the folds where the crossentropy performs better than our approach. Latent Dirichlet Attribution plus Hellinger distance performs better than our approach only in a 5.5% fraction of the cases. The authors of [89] report an overall 89% score with CE using leave-one-out validation. This score agrees with our reimplementation of the method and the ten-fold cross-validation. Our approach proves the best on all measures and all corpora.

When applying our method to the Email corpus, we can compare our results with those obtained by other authors since its deployment in 2011 for the PAN Authorship Attribution contest (during the CLEF 2011 Conference on Multilingual and Multimodal Information Access Evaluation). On the other hand, given the shortness of texts in this corpus, we cannot apply the crossentropy method. This method is based on compression, we excluded it from the comparison on the Email and the Blog corpora as we excluded the use of *LZ77 sequences*.

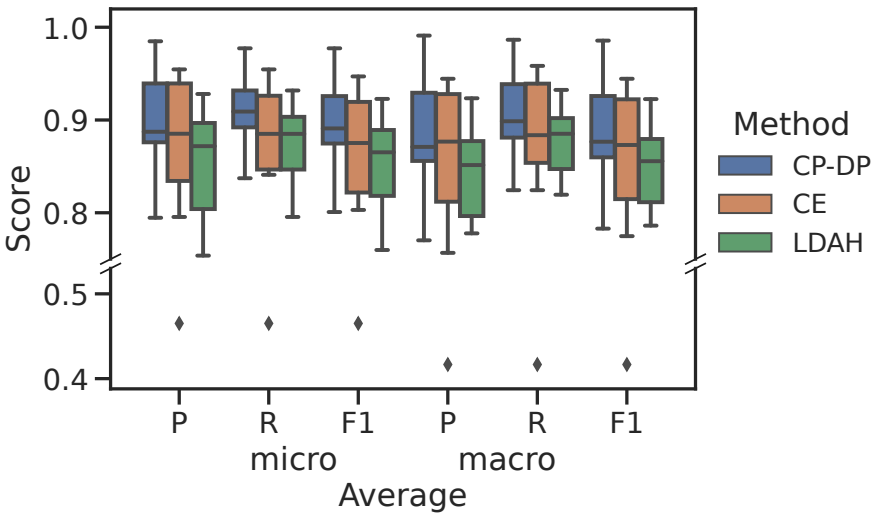


Figure 7.1. Comparisons of the different attribution procedures on the literary English corpus. Precision (P), recall (R) and F1 scores both micro and macro averaged, using the method detailed in this thesis (CP-DP), the Cross Entropy (CE) and the Latent Dirichlet Allocation plus Hellinger distance (LDAH).

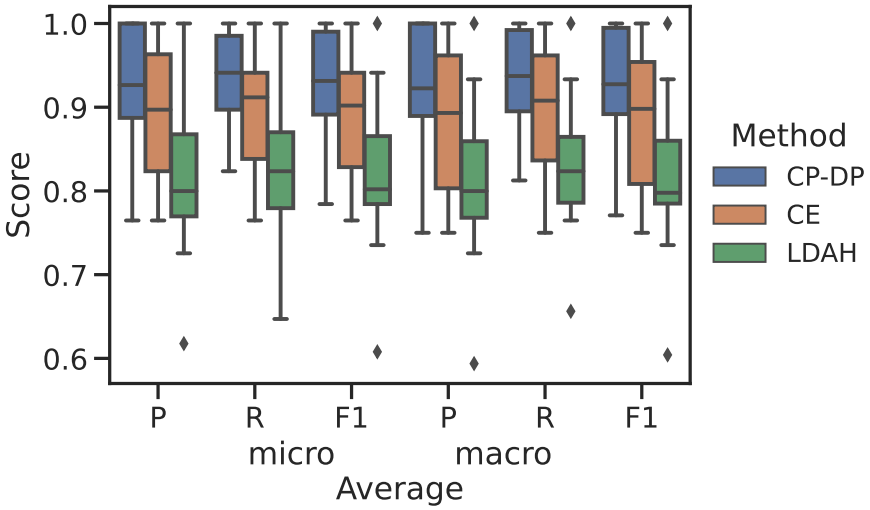


Figure 7.2. Comparisons of the different attribution procedures on the literary Italian corpus. Precision (P), recall (R) and F1 scores both micro and macro averaged, using the method detailed in this thesis (CP-DP), the Cross Entropy (CE) and the Latent Dirichlet Attribution plus Hellinger distance (LDAH).

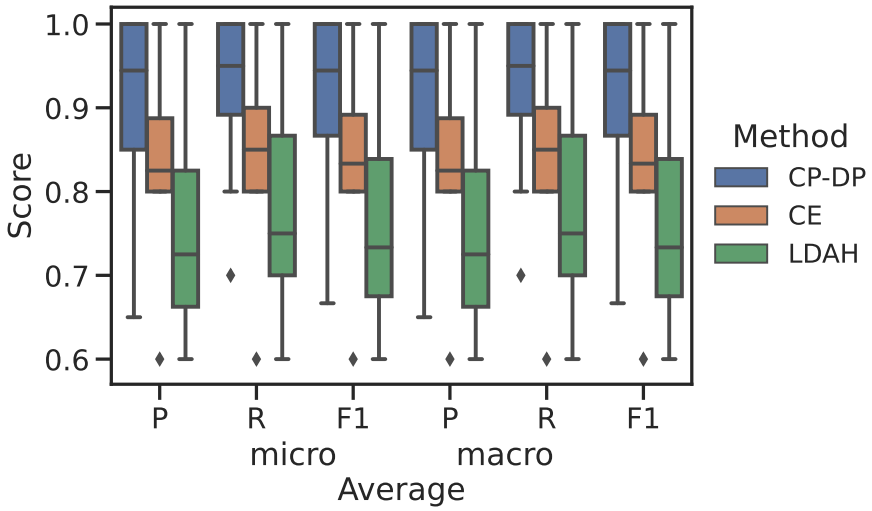


Figure 7.3. Comparisons of the different attribution procedures on the literary Polish corpus. Precision (P), recall (R) and F1 scores both micro and macro averaged, using the method detailed in this thesis (CP-DP), the Cross Entropy (CE) and the Latent Dirichlet Attribution plus Hellinger distance (LDAH). In this case, due to the reduced corpus size, each fold has only 10 (or even 9) unknown texts. It is of course easier to get all the attributions right on a single fold.

Table 7.1. Comparisons of the different attribution procedures on the literary corpora. The superscripts indicate the interquartile range (IQR) over the different folds: * $IQR < 5\%$, no superscript $IQR \in 5-10\%$, † $IQR \in 10-15\%$, ‡ $IQR > 15\%$. With a decreasing number of documents, the variation across de folds increases from the English to the Polish corpus.

Corpus	Method	Micro average			Macro average		
		Precision	Recall	F1	Precision	Recall	F1
English	CP-DP	0.901	0.913*	0.898	0.887	0.908	0.890
	CE	0.848†	0.853	0.840†	0.830†	0.850	0.830†
	LDAH	0.857†	0.877	0.856	0.845	0.879	0.852
Italian	CP-DP	0.928†	0.935	0.929†	0.920†	0.935†	0.928†
	CE	0.888‡	0.900†	0.890†	0.882‡	0.897†	0.886‡
	LDAH	0.814†	0.830†	0.815	0.808†	0.829†	0.813†
Polish	CP-DP	0.899‡	0.919†	0.906†	0.899‡	0.919†	0.900†
	CE	0.840	0.870†	0.843†	0.840†	0.850†	0.843†
	LDAH	0.748‡	0.769‡	0.755‡	0.748‡	0.769‡	0.755‡

The comparison for the Email corpus has the second role – validating our implementation of the Latent Dirichlet Attribution plus Hellinger distance method. Indeed, this approach was used on this corpus for the first time by Seroussi [133]) and later by Yang [157]. The difference between our results and those obtained by Seroussi and collaborators in 2012 is only 0.8%. The implementation of LDAH used by Yang yields even better results (Accuracy 0.48, 5.4% higher than the result from Seroussi). The results following this approach seem to depend on the implementation. Thus, we consider the discrepancy between the original implementation and ours acceptable.

The curators of the PAN workshop 2011 defined the test set for the Email corpus. To compare with other methods, in this case, we do not cross-validate our results and rely on the test set. Kourtis and Stamataos's approach is consistently the best on all measures, followed by DADT-P. We present our results and comparison with other methods in table 7.2.

The approaches reported in table 7.2 are related to ours as they use lexical features, N-grams [84] or word tokens [133, 135, 157]. Moreover, the last three are all members of the family of Topic Models. The approach of Kourtis and Stamatatos [84] obtained the best overall performance during the PAN'11 contest that introduced the corpus. We presented this approach in section 2.3.6 as the cooperation of two ML algorithms. We do not exclude the possibility of improving our results by joining forces with another classifier. The results obtained by Seroussi and collaborators in 2012 and 2014 use their implementation of LDAH from [134] and the evolution of the approach, the Disjoint Author-Document Topic Model [133], in the probabilistic version introduced in [135], see section 2.3.4. TDM is the Topic Drift Model introduced in [156].

The ranking of the best three approaches that emerged from the Email corpus test is inverted when applied to the larger Blog corpus. In table 7.3, DADT-P shows a worse capability to scale despite the authors considering the *a priori* probability of an author (it is χ_a, the corpus author distribution). In this case, the test set has the same imbalance as the training set. Using the probability of the authors is an advantage. We do not include any distribution on the authors. The four hyperparameters of our model must mediate any possible advantage from the imbalance of the corpus. TDM uses similarity and, unless there are hidden biases, it gets no advantage from the corpus imbalance.

Our approach proves to be considerably better than the others on these larger corpora with a relative difference of almost 20% from TDM. Moreover, our approach has the best ability to scale. It has the smallest relative and absolute loss in recall among the three methods tested on both versions of the corpus.

Table 7.2. Comparisons of the different attribution procedures on the Email corpus. The results obtained by Seroussi and collaborators in 2012 and 2014 use their implementation of LDAH from [134] and DADT-P [135]. Yang and collaborators obtained their result in 2017 using TDM [157]. The results of the PAN'11 workshop included all the six measures we are considering. The authors of the other three methods report only the percentage of test texts correctly attributed to their author, i.e. micro averaged recall.

Method	Micro average			Macro average		
	Prec.	Recall	F1	Prec.	Recall	F1
CP-DP	0.597	0.556	0.545	0.509	0.413	0.420
LDAH	0.594	0.418	0.469	0.436	0.347	0.362
Kourtis 2011 [84]	0.658	0.658	0.658	0.549	0.532	0.520
Seroussi 2012 LDAH [133]	–	0.426	–	–	–	–
Seroussi 2014 DADT-P [135]	–	0.594	–	–	–	–
Yang 2017 TDM [157]	–	0.542	–	–	–	–

Table 7.3. Comparisons of the different attribution procedures on the Blog corpus. The results obtained by Seroussi and collaborators in 2012 and 2014 use their implementation of LDAH from [134] and DADT-P [135]. Yang and collaborators obtained their result in 2017 using TDM [157]. The authors report only the percentage of test texts correctly attributed to their author, i.e. micro averaged recall.

Method	Prolific Recall	All authors Recall
CP-DP	**0.495**	**0.375**
Seroussi 2012 LDAH [133]	0.216	0.079
Seroussi 2014 DADT-P [135]	0.437	0.286
Yang 2017 TDM [157]	–	0.308

Table 7.4. Comparisons of the different attribution procedures on the informal corpora. The superscripts indicate the interquartile range (IQR) over the different folds: no superscript $IQR < 1\%$, † $IQR \in 1 - 2\%$. The Email corpus has train and test corpus already separated and we got no statistics on the validation folds.

Corpus		Method	Micro average			Macro average		
			Precision	Recall	F1	Precision	Recall	F1
Email		CP-DP	0.597	0.556	0.545	0.509	0.413	0.420
		LDAH	0.594	0.418	0.469	0.436	0.347	0.362
Blog	Prolific	CP-DP	$0.529^{\dagger}$	0.495	$0.493^{\dagger}$	$0.525^{\dagger}$	$0.567^{\dagger}$	$0.523^{\dagger}$
	All	CP-DP	0.442	0.374	0.375	0.333	0.300	0.313

Chapter 8

Computational Challenges

The implementation of this approach requires some care for the technical details. As mentioned in section 2.4.2, one of the drawbacks of a profile-based approach is its slowness in attribution. In our case, the training phase is extremely fast as it includes only the feature extraction end the optimisation of concentration and discount parameters. Then, we need to compare each document with every author's profile during attribution.

This comparison can be expensive. Following the approach described in chapter 3.1, for every fragment in the test corpus, we need to:

1. find the tokens already present in the reference author's profile;
2. compute the term from Eq. (3.2) for the tokens already present;
3. retrieve the base probability for the missing tokens;
4. compute the term from Eq. (3.2) for the missing tokens.

These four steps are repeated 13×10^9 times for the complete Blog corpus[1]. In this corpus, we must check about 10^{13} times if a token is in the author's process. In 3×10^{12} cases, we need the base probability. Without the appropriate precautions, all four passages can slow down the computation.

To save time on passages (1) and (3), we need to quickly retrieve the tokens' frequencies. The most efficient way to store and retrieve the tokens is by directly storing their hash. In this representation, we store every fragment as a collection of hashes with their multiplicities. This format allows using time-efficient hash maps saving the time of repeated hashing. The common representation of hashes as 8-byte integers is also memory efficient when using N-grams with $N \geq 9$ or variable width tokens.

[1] This is using the fragment size that yields the best results. We use almost all the posts (99.5%) at once. We split only those with more than 2800 tokens.

To speed up passages (2) and (4), we used the expression for the *Pochhammer (k-)symbol* from section 1.2.2. This allows to leverage on the available fast implementations of the Gamma function.

We found two other aspects to impact the running time substantially. First, the estimate of the concentration and discount parameters. Second, the size of the output files with the probabilities of every fragment against every author.

The estimate of the parameters is needed only once per author. However, a cross-validated experiment with many authors means 10^4 optimisations in the prolific authors' subset of the Blog corpus and 2×10^5 for the complete one. We already introduced the solution to this problem in section 3.3. The careful choice of the algorithm allows for fast convergence, and we used a fast sixth-order expansion of the Digamma function.

The second problem is tricky, affecting the running time at different levels. Indeed, large files are a problem of disk space and speed due to the hierarchical structure of modern computers' memory. Big objects are stored in slow memory far from the processors. We minimised the inconvenience by avoiding duplicate information, storing data in byte format and using tight-packed data structures. These precautions still imply about 250 GB of results for the complete Blog corpus. However, we decided to exclude data compression due to the considerable time to save about 15% of the size.

We parallelised most sections of the code to take advantage of our machine. We used 64 bits Dell cluster with two sockets. Each socket has ten 2.5 GHz multithreading cores. The system has 191 GB of memory.

The core part of the code is in C++, leaving only high-level operations to Python. For example, we used Python for text normalisation and punctuation removal due to the ease of string manipulation. On the other hand, the N-grams splitting and the LZ77-like tokenisations are in C++.

All this effort in optimisation was essential to keep the running time within reasonable limits. Given a set of hyperparameters (choice of variables, fragments size, P_0 normalisation), a typical cross-validated experiment on the literary corpora takes less than half minute, including optimising δ. For the complete Blog corpus, this time scales to about 40 hours.

To evaluate how the running time scales, we fix all hyperparameters except δ and measure the running time across corpora without splitting texts (infinite fragments size, single normalisation P_0, 5-grams). Our running time growth is in good agreement with a power-law of the total number of comparisons, see Figure 8.1, as expected for profile-based approaches. We observe an approximate linear growth if considering only informal corpora. Including also the literary corpora would imply

an exponent ~ 0.6 that has no explanation. For the literary corpora we observe an inflated running time due to overheads and longer texts.

The above relationship is true when using full documents. However, when using fragments, the time dependence is less straightforward. In figure 8.2, we present the running time for the English literary corpus with the sizes of fragments varying from the entire book to only three tokens. A power-law relationship of the running time with the number of fragments [2] holds only for small fragments, from ten to three tokens. When using fewer, bigger fragments, this growth is masked by a significant overhead that grows as the logarithm of the number of fragments. We ascribe this overhead to operations different from the comparison (preprocessing, attribution, store and load) that have a less direct dependency on the number of fragments. For example, the first time we compare a fragment with an author, the author's corpus is loaded in fast memory. This can take a long time compared with a single evaluation of the probability.

Is the discrete version practical? The scaling law in figure 8.1 clarifies the stress we put on keeping the approach simple and avoiding the need to sample the t_j. The total number of comparisons represents the leading term for the computing time growth. Indeed, the single comparison is quite fast in its optimised form. The leading term in the single comparison depends on the number of different tokens. This is the number of checks needed to know which tokens are missing in the reference author and the number of pairs of Gamma functions required for the probability of the sequence.

If we were to use the form for the PD process with discrete base probability, we would have sampled, for each token and not for every different one, if it was drawn from the P_0 or not[3]. Evaluating the marginal probability that the token was extracted from P_0 has a complexity $O(n_j)$. Even ignoring the time spent sampling the t_j in the author corpus, every Montecarlo step of an unknown text will take $O\left(\sum_{j=1}^{k} n_j^2\right)$. Multiply this for the number of MC steps, and the 40 hours needed for the attribution of the Blog corpus become weeks. To use different, faster sampling schemes, we would have needed to part from the intuition behind the CP-DP approach.

Text encoding We encoded all text with single-byte encodings. Using a single byte encoding for the texts simplifies working with the LZ77 compressor. This choice ensures that the compression algorithm does not split any letter over different tokens.

[2] The number of fragments is proportional to the number of comparisons and the inverse fragment size.

[3] More detail on the sampling using 'table indicators' in [23].

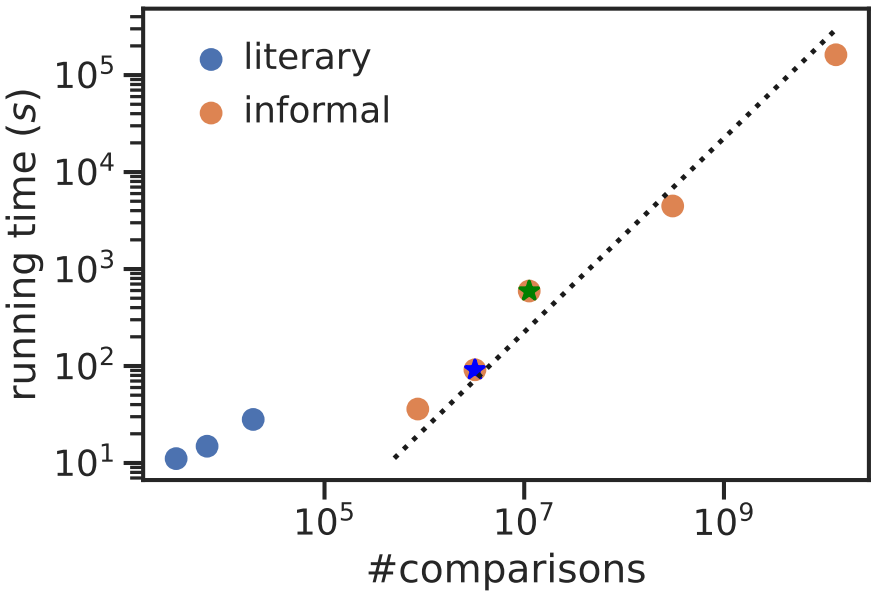


Figure 8.1. Running time in seconds as a function of the total number of comparisons using full documents. The number of comparisons is #authors × #documents. The point marked with a blue star is from the 100 most prolific authors in the Blog corpus. The point marked with a green star is from 100 random selected authors among the 1000 most prolific in the Blog corpus. The dotted line is a power-law function with exponent 1 as a guide for the eye.

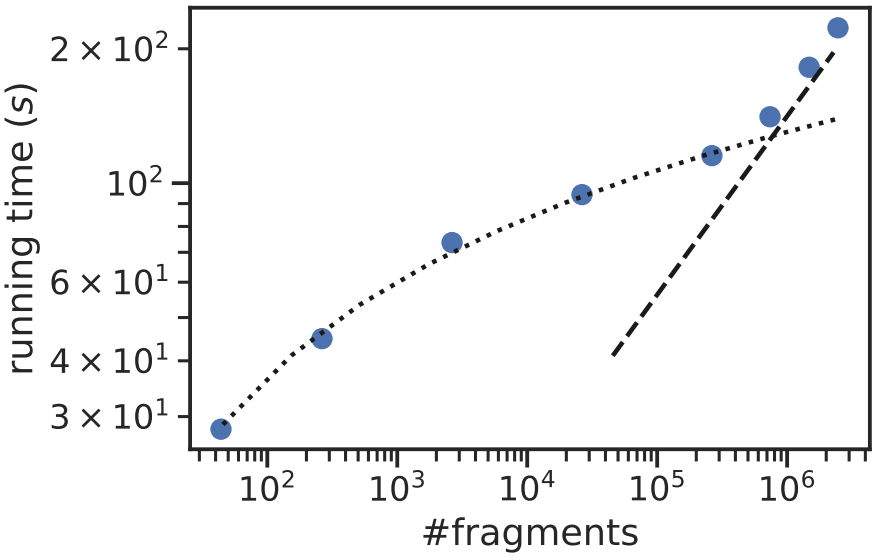


Figure 8.2. Running time in seconds as a function of the number of fragments. The dotted line is a logarithmic law. The dashed line is a power law with exponent 0.4. The lines are provided as guides for the eye.

At the same time, it maintains its simplicity in working with byte variables. Future versions of the code will include workarounds with variable size encodings such as UTF-8. Using a single-byte encoding is not required in any other code section. However, we used texts mainly written in languages using alphabets derived from the Latin that already have handy single-byte encodings: ISO 8859-1 (Latin-1) for English and Italian texts, ISO 8859-2 (Latin-2) for Polish texts.

Of course, the texts may contain quotes from other languages using different writing systems (from ancient Greek to Korean). However, these are rare enough, and we decided to transliterate those bits to their closest Latin representation.

Chapter 9

Possible Threats to Privacy

The analysis presented in this chapter stem from the ethical remarks in section 2.1.5. In recent years, we saw companies selling services to governments to spy on their citizens and people abroad. The last and most frightening example is the Pegasus scandal[1].

In this context, would a company like NSO be interested in offering software incorporating our algorithm to its clients? Would an authoritarian government ask for this kind of service? Similar reasoning is not new as Narayanan and collaborators [107] already in 2012 applied *authorship attribution* to a vast pool of authors with the idea of a deanonimisation attack. We will now try to present the setting where our algorithm may come in handy.

Setting We consider the case where a government finds some blog post dangerous and seeks to identify its author. We are not allowed any assumption on the topic, which might be anything from mentioning a more than thirty-year-old riot to some specific accusation against government members. We only assume that the post should be at least 50 characters long to be relevant. This is about the length of an average sentence. We ignore shorter posts or posts containing only the URL to some other website.

The stylometric analysis might be an option in this setting if the investigators established that the author is not among those already under strict surveillance whose actions are known. Also, the author effectively concealed the IP, and no useful information came from the OS/browser used or the post's metadata. Someone reminds a company offering stylometric analysis specifically for this purpose. If the author has ever posted anything under their real name, it might be possible to establish a link to the new evidence. Is it worth it?

[1] https://www.theguardian.com/world/2021/jul/18/revealed-leak-uncovers-global-abuse-of-cyber-surveillance-weapon-nso-group-pegasus, last checked January 12, 2022.

Methods We considered the complete Blog corpus as the one assembled likely to contain the author of the post. We considered all the posts for training and only posts at least 50 characters long for the test. In this case, we do not require that the actual author appears as the most probable. We only require that the actual author appears among the first k candidates. Then the police will put them under strict surveillance, waiting for the author to post again, or maybe they will abduct all the candidates on the list. We used ten-fold stratified cross-validation – and the hyperparameters set – that gave the best results in chapter 7.

This analysis takes inspiration from the work of Narayanan and collaborators "On the Feasibility of Internet-Scale Author Identification" [107]. We want to remark on a few differences in the methods. First, the number of candidate authors used in [107] is more than five times the number of authors in the Blog corpus. This difference is a clear advantage for us. However, all the blogs considered in [107] contained at least 7500 characters. A similar condition would force us to discard almost 40% of the blogs. Therefore, we kept all blogs, and the presence of the smaller ones is a penalty. Second, they removed non-English language blogs. Third, in [107], they used three posts for each blog. In this case, instead, we attempt to identify the author with a single sample.

Results We report in figure 9.1 the results varying the size k of the shortlist. We get the actual author among the first 20 for 62.5% of the posts without any further processing of the results. Narayanan and collaborators obtained 20% of the posts to have their author in the top 20. By introducing confidence estimation, they further improved the results, reaching 80% of posts with shortlisted authors halving the recall. We leave the study of the improvements with confidence estimation to future work.

These results share many of the limitations already pointed out in [107]. We did not consider active *obfuscation* or attempted cross-domain identification (e.g., finding the blog post's author having only email samples). However, given the current state of *authorship obfuscation*, these results suggest risk in at least one domain.

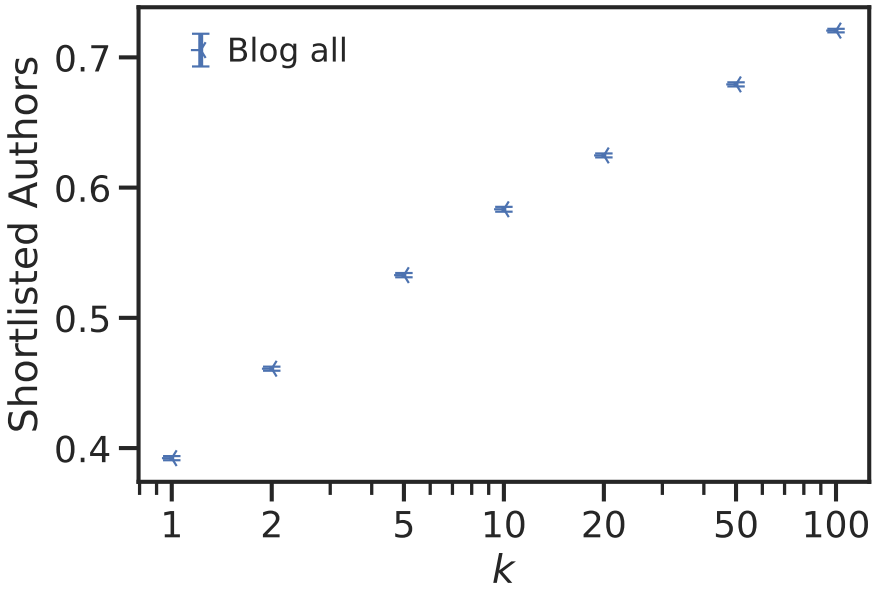


Figure 9.1. Fraction of post whose author is correctly shortlisted. An author is correctly shortlisted if their id appears in the first k most likely.

Conclusions

The method we proposed in this thesis is relevant for its outstanding results and the simplicity of its approach. First of all, we reached the best performance on record on the most extensive and challenging corpus. One of the greatest challenges for most approaches is scalability. In our case, even increasing the number of documents and candidates by three orders of magnitude we retain two-fifths of the correct attributions. Second, we successfully overcame the strong imbalances in the informal corpora. The style samples for every author were highly varied in number and size. This kind of corpora can disrupt many other approaches. Third, we proved that it is possible to apply this approach to different languages without losing performance.

At the core of our approach lies the intuition of modelling each author as a stochastic process. Therefore, we chose a PD process for its superior ability in representing natural systems. Furthermore, this choice provides us with a straightforward way to estimate the likelihood of every author. We used the probability of the unknown text conditioned to the PD process and the author's production.

Besides the practical results, we showed the role and meaning of the constituents of the model. We introduced three possible kinds of variables for the representation of the documents. We discussed their specific characteristics and the conditions where they perform well or are unusable.

We discussed the effects of our assumption of a discrete base probability distribution. This assumption is equivalent to fixing the number of extractions of each element from the base probability to one. This imposition required some considerations about the form of the base probability and the introduction of a parameter compensating the missing information.

We discussed the opportunity of dividing texts and authors into smaller chunks. This step is needed to cope with unbalances in the corpora. In particular, we showed that, taking the limit of single token fragments, our approach reduces to a variant of methods based on Kullback-Leibler divergence. Furthermore, in the opposite limit of long fragments, we identified a bias towards authors with a small corpus. We discussed the origin of this bias and how it changes direction, varying the ratio between the fragment and author lengths.

Given the limitations of our setting, we showed how our approach already presents privacy concerns as it is, without any optimisation towards this specific task. Moreover, with the lowering cost of surveillance, reducing the set of candidates by three orders of magnitude represents a concrete menace to maintaining anonymity online.

This kind of software should be regulated as it is already happening with facial recognition algorithms. However, this may not be enough if we parallel with facial recognition. For example, a company like the US-based Clearview – reporting the collaboration with police forces across the USA on its homepage[2] – has not changed its business model after many condemnations received in at least five countries and three continents[3].

The need for effective *authorship obfuscation* software may quickly become a matter of preserving democracy and human rights. Therefore, *authorship obfuscation* software should become as available and usable as typical spell checkers, overcome the present problems with "preserving semantics", and undergo systematic tests against all the latest *authorship attribution* techniques.

The future of this approach is manifold. For example, we may apply it to different tasks like author profiling in stylometry. Alternatively, we could test its effectiveness on challenging genres such as poetry or text in non-Indoeuropean languages. Indeed, we applied our approach to three European branches of the Indoeuropean family (Romance, Germanic and Slavic), all with alphabetic writing. On the other hand, its effectiveness on texts in languages like Amharic, Arab, or Chinese is not proven yet.

Our approach may find application in biology too. There has been a technical and conceptual exchange between linguistics and genetics in the last decades. Concepts like Authorship Attribution were introduced in biology, for example, to find the origin of "genome segments thought to arise by horizontal transmission between species" [132]. At the same time, the modelling of nucleotide sequences as the output of stochastic processes has been investigated since the nineties to characterise the type of RNA (messenger, ribosomal, transfer) [85], and compression approaches similar to the cross-entropy method have been proposed [30, 74].

We can envision applications of our technique in this field. For example, we may tackle RNA classification problems by leveraging the ease of deriving grammar trees, e.g. from RNA sequences, representing both the primary and the secondary structure [127, 91]. These representations are unambiguous, and RNA, unlike texts,

[2] https://www.clearview.ai/, last checked April 14, 2022.

[3] See for example https://www.theguardian.com/world/2021/nov/03/facial-recognition-firm-cleaview-ai-to-appeal-order-to-stop-collecting-images-of-australians or https://www.garanteprivacy.it/home/docweb/-/docweb-display/docweb/9751323#english for Australia and Italy respectively, last checked April 14, 2022.

does not require PoS tagging. Using a linear representation of the grammar tree as the input sequence may encode more information for our approach to exploit.

We also envision possible evolutions of the approach itself. For example, reintroduction of a proper discrete base probability exploring other ways to make this version faster and better in performance. Also, introducing a hierarchy could open interesting perspectives, provided that the slowdown is made sustainable.

A different path is changing the model itself. For example, we could use models like the PUT model and its extensions. They share some features with the PD process. However, these models allow going beyond exchangeability. Introducing semantic and temporal correlation in our model would help represent textual data closely with possible performance improvements.

To extend the possibilities of our approach beyond our imagination, we plan to release the CP-DP soon as open-source software.

Appendix A

Crossentropy

The concept of crossentropy came in multiple times in the main text. Crossentropy may be defined as the extra effort needed to transmit a message from a source using a coding system optimised on a different source. Here, a source is intended in the information theory sense as any process that generates successive messages. All sources are stochastic and, for simplicity of mathematical treatment, two classes often considered are zero-memory and stationary ergodic sources. The first emit i.i.d. random variables while the second impose less restrictive constraints.

We clarify this with a historical example: the Morse code. In Morse code any message is transmitted as series of dots and dashes (three times the length of a dot) interleaved with breaks of different length. The message can be transmitted through the flashes of a light, the sound of a buzzer, a pen writing on a running strip of paper, in any case the number of dots and dashes determines the duration (and the cost) of the message.

Samuel Morse devised the code that goes under his name[1] so that the most common English letter '`e`' is coded with a single dot while the less common '`j`' requires one dot and three dashes. Transmitting an Italian text with the Morse code optimised for English will result in a longer message than using a coding optimised for Italian. For example the five most common letters in English are "`e t a o n`" while the most common letters in Italian are "`a e i o n`". This would lead to some loss in performance for the extra time spent for '`a`' and '`i`' while reserving a fast code for the '`t`' that is only the seventh most common[2]. The crossentropy between Italian and English in this case would be given by the average extra time (i.e. bits)

[1] The idea behind the code we know is due to his assistant Alfred Vail. Morse's idea was to code entire words. This could achieve higher performance on specific texts (e.g. commercial, military, ...) but is impractical for humans (see Egyptian hieroglyphs).

[2] In the International Morse Code the shortest letters are "`e t i a n`" while '`o`' has one of the longest codes with three dashes. This is because it was created as Continental Morse Code in a European conference and tends to accommodate different needs.

per character needed to code a message in Italian using the Morse code optimised for English.

In a more formal way, let us consider two stationary zero-memory sources A and B emitting sequences of 0 and 1. A emits a 0 with probability p and 1 with probability $1-p$ while B emits 0 with probability q and 1 with probability $1-q$. The Shannon entropy of source A is defined as:

$$H(A) = -p\log_2 p - (1-p)\log_2(1-p) \tag{A.1}$$

An ideal compression algorithm[3], applied to a sequence emitted by A will, encode the sequence [155], using on average a number of bits per character equal to the Shannon entropy $H(A)$ of the source.

The adopted coding optimal for source A, will have a worse performance on the sequence emitted by B. For example, if $p > q$ we will use few bits ($-\log p < -\log q$) to encode the 0s, but this happens only a fraction $q < p$ of the time. The relatively more common 1s will take $-\log(1-p) > -\log(1-q)$ bits. The number of bits per character needed to encode the sequence emitted by B in the coding optimal for A will be

$$H(q,p) = -q\log_2 p - (1-q)\log_2(1-p) \tag{A.2}$$

while the entropy per character of the sequence emitted by B in its own optimal coding is $H(q,q) = H(B) = -q\log_2 q - (1-q)\log_2(1-q)$. Notice $H(q,p)$ is always greater than $H(q,q)$ and $H(q,p)$ approaches $H(q,q)$ from above when q approaches p.

The two sources may emit sequences of symbols from alphabets $\mathcal{X}$ of any size. As in the example, the crossentropy still measures the number of bits needed to encode messages from one of the sources when using the optimal code for the other. The general formula for two sources represented by their probability distributions P and Q over the symbols $x \in \mathcal{X}$ will be:

$$H(P,Q) = -\sum_{x\in\mathcal{X}} P(x)\log Q(x) \tag{A.3}$$

Notably the crossentropy is related to Kullback–Lebler divergence as:

$$H(P,Q) = H(P) + D_{KL}(P \parallel Q) \tag{A.4}$$

which is defined as $D_{KL}(P \parallel Q) = \sum_{x\in\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)}$ and is a common measure of the difference of P with respect to Q used also in stylometry.

[3]Or an asymptotically optimal compression algorithm, like LZ77, in the limit of an infinite sequence.

As noted in the main text, the crossentropy[4] is not a measure of distance from a mathematical point of view. While it's positive semidefinite (being zero only when $P = Q$ and $H(P) = 0$), in general is not symmetric ($H(P, Q) \neq H(Q, P)$) nor satisfies the triangular inequality ($H(P, Q) \lneqq H(P, R) + H(R, Q)$). Even if it is possible to enforce, in a particular setting, see [15, 12], its behaviour as a distance this is not of general interest or usefulness.

[4] As the Kullback–Leibler divergence.

Appendix B

Alternative Definitions

For completeness, we present in this appendix a selection of alternate definitions we considered for some aspects of our model. We excluded these alternatives for performance or consistency reasons. The following sections may answer why we did not follow some other approach.

B.1 Alternatives in Preprocessing

In section 2.2.1 we mentioned how often a step in the preprocessing phase is case unification. This means converting all the letters to lower- (or equivalently, upper-)case to avoid the creation of spurious tokens. The same word appearing at the beginning or in the middle of a sentence will then define a single token.

In our approach, we do not unify the case of letters as we observed a reduced performance for informal corpora and mixed results for the literary.

We report in table B.1 the final results of the cross-validation on blogs using *OSF N-grams* obtained with and without case unification. In the case of *Dictionary words*, we decreased performance too. For the Blog corpus – prolific authors, the score decreases from 0.483 to 0.471.

In table B.2, we present a comparison of the results with and without case unification for the literary corpora. In this case, the picture is less clear. Changing corpus and kind of features the results may benefit from case unification. However, in general, the results are worse or unchanged.

B.2 Alternative Fragments

In the main text, we always considered extracting the features from full texts and only later split the sequences of tokens into fragments. A different approach is to split the original texts before tokenisation. We chose a suitable length, and then we

Table B.1. Effect of case unification on the best hyperparameters' choice for the Blog corpora. The superscripts indicate the interquartile range (IQR) over the different folds: no superscript $IQR < 1\%$, † $IQR \in 1 - 2\%$.

Corpus		Unified case	Micro average			Macro average		
			Precision	Recall	F1	Precision	Recall	F1
Blog	Prolific	No	0.529†	0.495	0.493†	0.525†	0.567†	0.523†
		Yes	0.515	0.478	0.480	0.509†	0.507	0.553†
	All	No	0.442	0.374	0.375	0.333	0.300	0.312
		Yes	0.424	0.358	0.361	0.319	0.287	0.300

Table B.2. Scores using MR attribution for the literary corpora. For each kind of feature we used an average of the hyperparameters that offered the best results in the attribution task.

Corpus		Unify case	
		No	Yes
ITA	DICT	**0.901**	0.895
	OSFG	0.906	**0.918**
	LZ77	**0.857**	0.842
ENG	DICT	0.900	**0.907**
	OSFG	0.929	**0.932**
	LZ77	**0.887**	0.886
POL	DICT	**0.869**	0.848
	OSFG	0.899	0.899
	LZ77	**0.929**	0.909

split the text into fragments that number of characters long. Each fragment then goes through feature extraction. To avoid the creation of spurious tokens is better to split at the closest space. However, the effect of the different sizes of the fragments is quickly negligible if the lengths are not too small[1]. This choice retains all the information about the different fragments. Some may be closer or further from their author's style; different attribution techniques may exploit these differences to improve the results. When using LZ77 sequences, this choice suggests looking for repeated sequences on a copy of the fragment itself instead of using a sliding window. We split the text into fragments of the same size as the LZ77 window and look for repeated sequences on a copy of the fragment. Since the sequences at the beginning of a fragment can always find matches with the end, this solution gives slightly different dictionaries. This is the version used in the crossentropy method we use to compare. Searching this way also avoids the bias towards the sequence's end of the *gzip* version.

Another approach is to tokenise the text and then split it into fragments, all with the same amount of tokens but this time sampling them at random. In this case, we assume the complete exchangeability of the words in the text. This choice is in line with the use of the PD process however is not valid in general, and a long text may exhibit style changes that the sampling would cancel. Furthermore, as we noted in chapter fragments, the size of the fragments may capture some correlation length. This implies accounting for medium-scale correlations in the text beyond exchangeability. We tested this second option briefly and observed worse performance.

Instead, we devoted some study to the first option during preliminary analysis. Looking at the curves in figure B.1, it seems that some features allow working with shorter fragments. For example, for fragments one hundred characters long, the attribution using LZ77 gives scores under 0.5. The other approaches seem yet unaffected. We noticed that the curves in the figure are more aligned when plotted as a function of the average number of tokens per fragment, see figure B.2. In this case, for fragments with a few tokens, the performance of all features drops as already observed, e.g. in the bottom panels of figures 6.1, C.12, and C.13.

We chose to keep the feature extraction separated from the division into fragments. We conclude that, due to the varied number of tokens per fragment, splitting before the feature extraction adds more noise than information.

[1]The median length of words in texts of the three languages considered (English, Italian and Polish) is below six characters. Therefore, most fragments would have a length within three characters from the designed size.

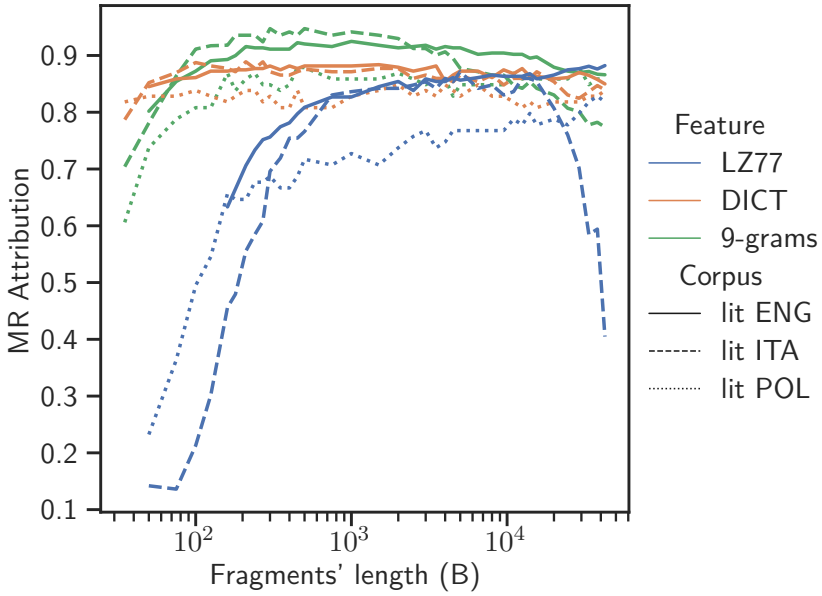


Figure B.1. Attribution varying the fragments' byte length.

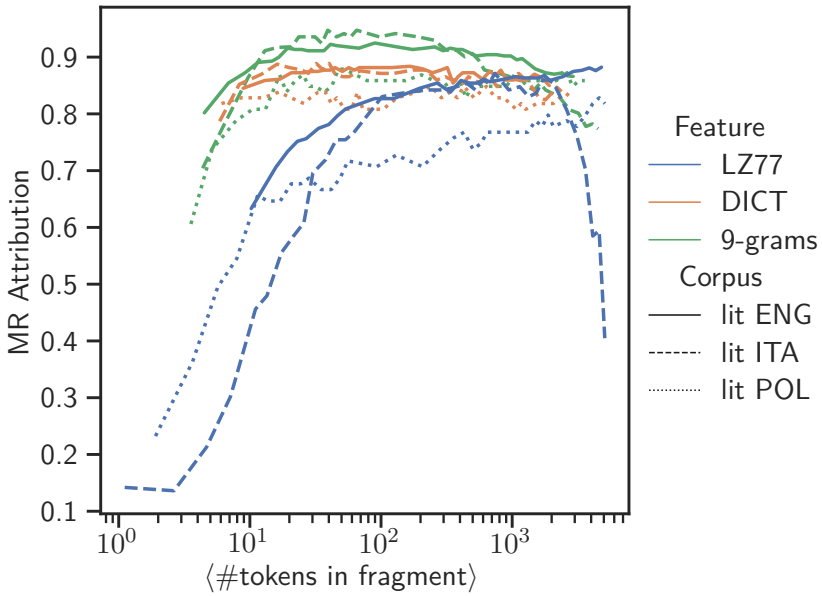


Figure B.2. Attribution varying the average fragments' length in tokens.

B.3 Alternatives for the Base Probability

In chapter 5 we decided to weight the tokens using their occurrences in the whole corpus. This is a quite straightforward choice but not the only one possible. Using the token frequency as a proxy for its probability does not take into account the reinforcement process in action in the PD process. On the other hand, it offers a fine grained distinction of weights that may span several orders of magnitude. Increasing the corpus size results in a better description of the distribution with more tokens included and a better estimate of the frequency for the most common ones.

A simpler, agnostic choice would be to assume all tokens having equal *a priori* probability. The different counts observed would derive only by the reinforcement active in the process. However, this perspective ignores the fact that the most common elements are shared across different texts and authors.

An different approach moves from the process itself. We assume that each author is associated to a single process, and each process can extract each token from the base distribution P_0 only once. Thus, we may get a proxy of the weight per token as the fraction of authors that uses it. This is practically equivalent to the trivial sampling from a sparse Dirichlet-Multinomial distribution as the identity and the number of extractions is fixed.

This approach gives an extremely coarse estimate limited by the number of authors. In all but one the corpora we considered the difference between the most and least probable tokens would be less than two orders of magnitude. This compression might be more noticeable on frequent tokens: every token appearing in (almost) every author, even if only once, would have the same probability as the most common ones. However, for this same reason, in most of the cases those would be tokens known to the author and their base probability would never be used.

In this case, increasing the corpus size with more examples from the same authors would give little improvement. This is the case when the set of possible authors is closed and we try to obtain a better estimate of the likelihoods. In many kinds of corpus the estimate will remain coarse grained with more words pushed to the upper limit of the weight while new ones enter.

An intermediate approach would be counting the number of texts in which every token is present. The number of texts is not smaller than that of the authors allowing for a finer distinction between different tokens even if forcing the definition of the process.

This option is strongly influenced by the kind of corpus. With literary texts might be very close to the author count, every author has (relatively) few long texts. With less formal writing like messages of few lines each, the estimate gets closer to the global occurrence count. In this method, like in the first one but unlike the

author count, the P_0 is independent from the attribution of texts. This is useful in practical situations when one wants to use the largest amount of texts and the P_0 to be independent from the validation fold.

In principle one could object that the right thing to do is to count the number of authors using every token. However, this is in contrast with the assumption of a continuous probability distribution and leads to very rough estimates, due to the often limited number of authors. It also poses problems on how to count the tokens of the books of unknown authorship, especially those that are unique to these books.

Counting the global number of occurrences or just the number of texts with at least one occurrence leads to smaller differences than one may expect. Fist of all it affects only tokens that appear more than once per text. If texts are short, as in the informal corpora, very few very common tokens appear more than once in most of the texts. All but the most common tokens are affected lightly by this choice[2]. Second, the tokens that are relevant when computing the conditional probabilities are in general low probability and high rank. Even in longer texts, as in the literary corpora, these tokens appear a few times per text. For all but the most frequent the effect is again small.

Looking at figure B.3 is easy to note how, for an informal corpus – bottom panel, the difference is mainly in the sum of the counts. This in turn affects the fixed normalisation but not the updated as the counts of higher rank tokens are affected slightly. For a literary corpus there is a reduction of the probability for low rank tokens but the less probable ones are less affected. For the Polish corpus in figure, the most frequent tokens have their probability depressed as the counts saturates for the most frequent tokens. As the corpus contains one hundred novels, no token may have more than one hundred counts, no matter how frequent. Even for ranks in the thousands, where we find tokens appearing in about 50 texts, we clearly notice effects of saturation.

The difference is limited to a specific class of tokens and almost only for literary corpora. It affects tokens with a rank in the order of the thousands for *Dictionary words* and *LZ77 sequences* with small windows, in the order of the tens of thousands for *OSF N-grams* and *LZ77 sequences* with large windows. In figures B.4 and B.5 I report the comparison of the results obtained with the P_0 used in the main text. The difference is often small and, in correspondence of the best values, in favour of the global count approach. For literary and informal corpora, the best results under different choices of P_0 are in tables C.4 and C.5.

[2]If working with N-grams many of the most common words are discarded as too short and the effect is even smaller.

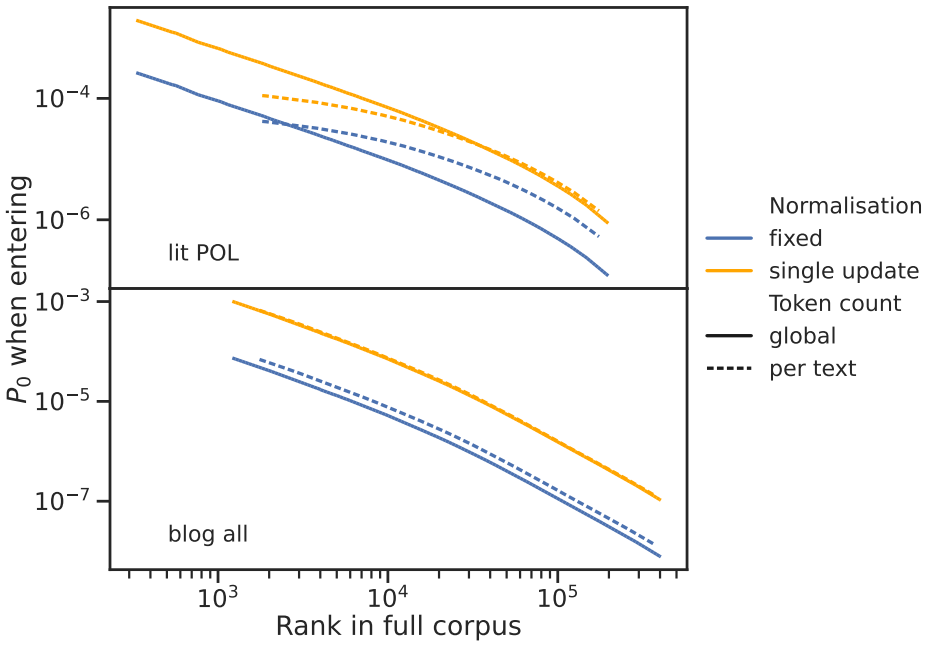


Figure B.3. Example of effective P_0 with different token counting. The lowest rank with the per-book counting of tokens is necessarily higher than with the overall counting. The probability of tokens with few overall occurrences that tend to appear in many texts increases. These tokens move to lower ranks but because widespread across texts there's a higher chance that are used also in one of the texts of the reference author. Points in this plot refer only to tokens that are absent in every text of the reference author.

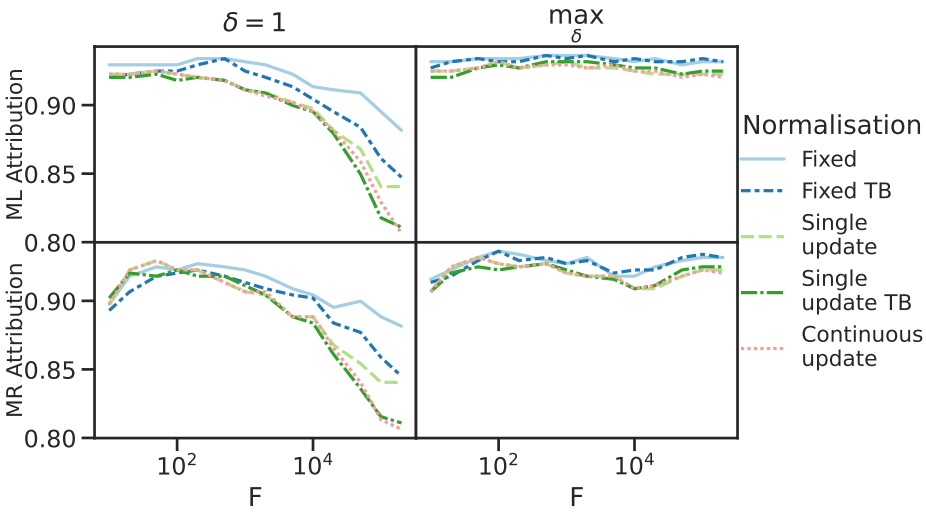


Figure B.4. Text attribution in the literary English corpus with the different choices of P_0. The normalisation curves marked as "TB" are Text Based, obtained counting the number of texts each token appears in. We report the best scores using the different choices of P_0 in table C.4.

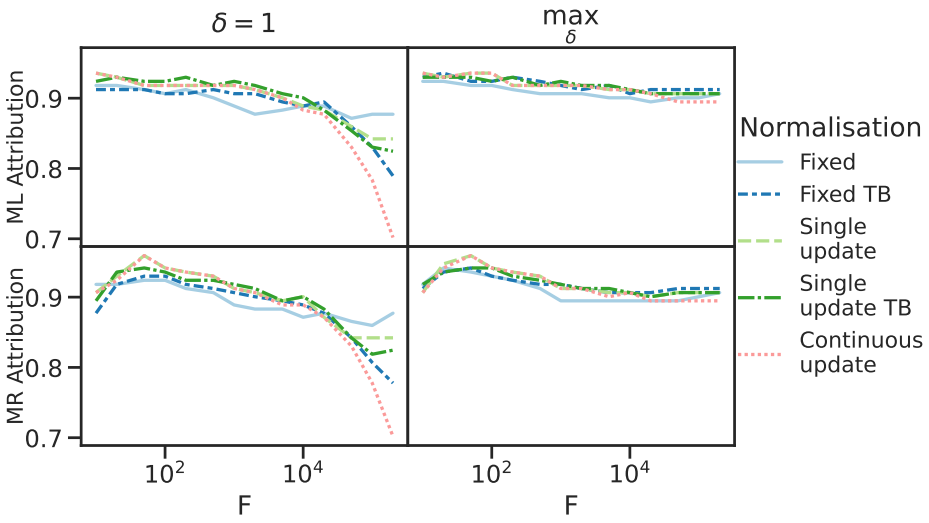


Figure B.5. Text attribution in the literary Italian corpus with the different choices of P_0. The normalisation curves marked as "TB" are Text Based, obtained counting the number of texts each token appears in. We report the best scores using the different choices of P_0 in table C.4.

Appendix C

Additional Graphs and Tables

This appendix includes graphs and tables useful for a deeper understanding of the subject of this theses that were excluded from the main text.

Table C.1. Best results using *Dictionary words*. In case of multiple choices of the hyperparameters giving the same results, we report the value of δ closest to 1 and the larger fragment size. In most cases, the best scores are obtained using *fixed normalisation* for the P_0. We report the best score in boldface if Maximum Likelihood and Majority Rule attribution give different results for the same corpus. "PRO" is the prolific authors subset of the Blog corpus.

	Corpus	attr.	$\delta = 1$		$\max_\delta$		
			F	Score	F	δ	Score
Literary	ENG	ML	10^4	**0.897**	10^4	0.40	0.902
		MR	10^3	0.893	10^3	0.32	0.902
	ITA	ML	10^4	0.860	10^3	2.00	0.889
		MR	100	**0.877**†	100	3.50	**0.918**
	POL	ML	10^5	**0.879**	10^4	2.50	0.899
		MR	full	0.869	10	4.00	0.899
Informal	Email	ML	10	**0.501**	10	0.25	**0.519**†
		MR	10^3	0.492	10^4	0.22	0.518†
	PRO	ML	full	0.485	500	1.80	0.487
		MR	full	0.485	full	1.80	0.487

† *single update normalisation* gives the best results.

Table C.2. Best results using *OSF N-grams*. In case of multiple choices of the hyperparameters giving the same results, we report the value of δ closest to 1 and the larger fragment size. In most cases, the best scores are obtained using *fixed normalisation* for the P_0. We report the best score in boldface if Maximum Likelihood and Majority Rule attribution give different results for the same corpus. "PRO" is the prolific authors subset of the Blog corpus.

	Corpus	attr.	$\delta = 1$			$\max_\delta$			
			F	N	Score	F	N	δ	Score
Literary	ENG	ML	500	8	**0.934**	2000	8	0.79	0.936
		MR	50	9	0.929†	100	10	0.20	0.936
	ITA	ML	10	9	0.936†	10	9	1.00	0.936†
		MR	50	10	**0.959**†	50	10	1.00	**0.959**†
	POL	ML	1000	9	0.909	200	10	0.79	0.919
		MR	500	10	**0.919**	100	10	1.30	**0.929**
Informal	Email	ML	10	3	**0.512**	20	4	0.25	**0.532**
		MR	full	3	0.502	full	4	0.13	0.530†
	PRO	ML	5000	5	0.493	5000	5	1.30	0.494
		MR	full	5	0.493	full	5	1.30	0.494

† *single update normalisation* gives the best results.

Table C.3. Best results using *LZ77* on literary corpora. In case of multiple choices of the hyperparameters giving the same results, we report the value of δ closest to 1 and the larger fragment size. In most cases, the best scores are obtained using *fixed normalisation* for the P_0. Using this kind of feature the MR attribution is consistently better and requires fragments about one thousand times smaller.

Corpus	attr.	$\delta = 1$			$\max_\delta$			
		F	L	Score	F	L	δ	Score
ENG	ML	5×10^4	5.6×10^4	0.888	2×10^3	5.6×10^4	1.1	0.891
	MR	20	1.0×10^5	**0.897**‡	20	5.6×10^4	1.4	**0.909**
ITA	ML	2×10^4	5.6×10^4	0.854†	1×10^4	3.2×10^4	1.1	0.871†
	MR	10	1.8×10^5	**0.906**†‡	10	1.8×10^5	1.0	**0.906**†‡
POL	ML	5×10^3	3.2×10^5	0.949†‡	1×10^3	5.6×10^4	1.1	0.949
	MR	20	3.2×10^5	**0.960**†	20	3.2×10^5	1.0	**0.960**†

† *single update normalisation* gives the best results.

‡ same best results also with shorter windows and longer fragments.

Table C.4. Best results with the different P_0 normalisations. The text-based (TB introduced in section B.3) weights or the continuous update normalisation often offer the best results. However, in every such case, the fixed normalisation or the single update also offers the same score.

Corpus	attr.	P_0	$\delta = 1$			$\max_\delta$			
			F	N/L	*score*	F	N/L	δ	*score*
ENG	FNN	Fixed	500	8	0.934	2.0×10^3	8	0.79	0.936
		Fixed TB	500	8	0.934	500	8	0.89	0.936
		Single	50	8	0.925	100	9	0.63	0.932
		Single TB	50	10	0.923	1.0×10^3	10	0.63	0.932
		Continuous	50	8	0.925	100	9	0.63	0.932
	MR	Fixed	200	8	0.927	100	10	0.20	0.936
		Fixed TB	200	8	0.923	100	10	0.32	0.936
		Single	50	9	0.929	50	9	0.79	0.932
		Single TB	100	10	0.923	500	8	0.63	0.927
		Continuous	50	9	0.929	50	9	0.79	0.932
ITA	FNN	Fixed	20	8	0.918	20	9	0.79	0.924
		Fixed TB	20	9	0.912	20	11	0.63	0.936
		Single	10	9	0.936	10	9	1.00	0.936
		Single TB	20	10	0.930	20	10	1.00	0.930
		Continuous	10	9	0.936	10	9	1.00	0.936
	MR	Fixed	100	8	0.924	20	10	0.71	0.942
		Fixed TB	100	8	0.930	50	9	0.71	0.942
		Single	50	10	0.959	50	10	1.00	0.959
		Single TB	50	10	0.942	50	10	1.00	0.942
		Continuous	50	10	0.959	50	10	1.00	0.959
POL	FNN	Fixed	2.0×10^4	1.0×10^5	0.939	1.0×10^3	5.6×10^4	1.10	0.949
		Fixed TB	2.0×10^4	1.0×10^5	0.939	1.0×10^5	1.0×10^5	0.89	0.949
		Single	5.0×10^3	3.2×10^5	0.949	5.0×10^3	3.2×10^5	1.00	0.949
		Single TB	5.0×10^3	3.2×10^5	0.949	5.0×10^3	3.2×10^5	1.00	0.949
		Continuous	5.0×10^3	3.2×10^5	0.949	5.0×10^3	3.2×10^5	1.00	0.949
	MR	Fixed	20	1.8×10^5	0.939	20	1.8×10^5	0.89	0.949
		Fixed TB	20	1.8×10^5	0.949	10	1.8×10^5	0.79	0.960
		Single	20	3.2×10^5	0.960	20	3.2×10^5	1.00	0.960
		Single TB	1.0×10^4	3.2×10^5	0.949	1.0×10^4	3.2×10^5	1.00	0.949
		Continuous	20	3.2×10^5	0.960	20	3.2×10^5	1.00	0.960

Table C.5. Best results with the different P_0 normalisations. The text-based (TB introduced in section B.3) weights or the continuous update normalisation often offer the best results. However, in every such case, the fixed normalisation or the single update also offers the same score. PRO is the prolific authors subset of the Blog corpus.

Corpus	attr.	P_0	$\delta = 1$			$\max_\delta$			
			F	N	*score*	F	N	δ	*score*
Email	FNN	Fixed	10	3	0.512	20	4	0.25	0.532
		Fixed TB	10	3	0.504	10	4	0.20	0.531
		Single	10	4	0.508	10	4	0.20	0.531
		Single TB	10	4	0.487	10	4	0.14	0.530
		Continuous	10	4	0.508	10	4	0.20	0.531
	MR	Fixed	full	3	0.502	full	4	0.16	0.529
		Fixed TB	1.0×10^3	3	0.491	full	4	0.11	0.529
		Single	2.0×10^3	4	0.491	full	4	0.13	0.530
		Single TB	5.0×10^3	4	0.467	full	4	0.14	0.530
		Continuous	2.0×10^3	4	0.485	1.0×10^4	4	0.14	0.530
PRO	FNN	Fixed	5.0×10^3	5	0.493	5.0×10^3	5	1.30	0.494
		Single	50	5	0.485	50	5	0.71	0.487
	MR	Fixed	full	5	0.493	full	5	1.30	0.494
		Single	full	4	0.479	full	4	0.45	0.485

Table C.6. Fitted values of the β exponent for the Heaps's law plot using *OSF N-grams*. N^{best} is the value that gives the best scores.

Corpus		$N = 3$	$N = N^{best}$	$N = 12$
Literary	Polish	0.373	0.694	0.714
	Italian	0.394	0.572	0.645
	English	0.270	0.684	0.807
Email		0.451	0.515	0.719
Blog	all authors	0.388	0.563	0.846
	prolific	0.350	0.531	0.856

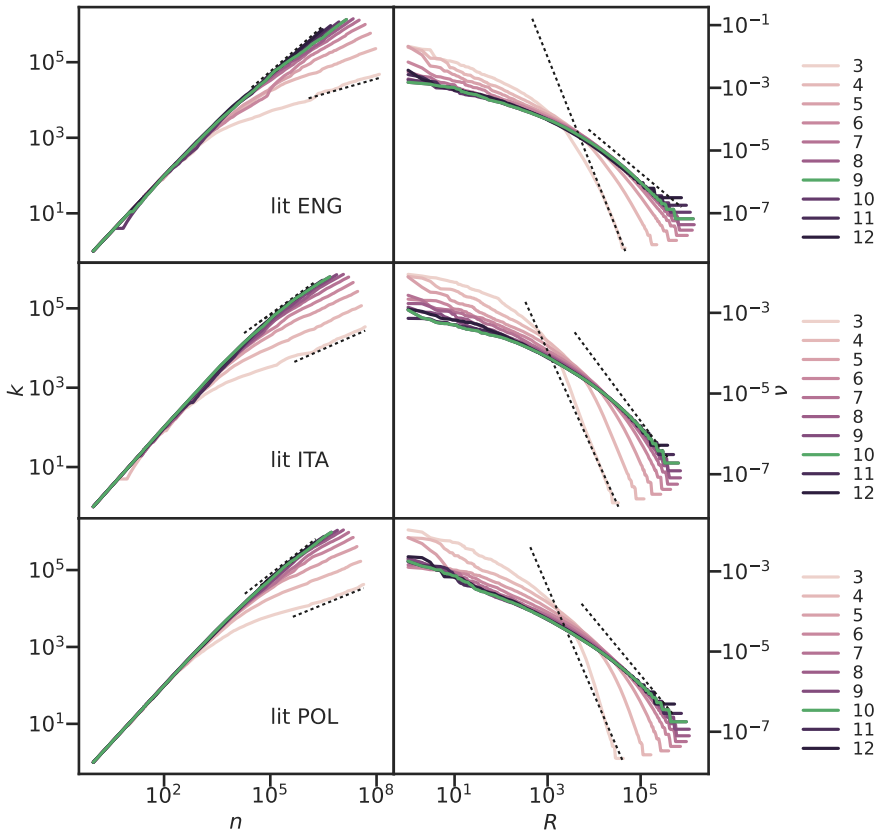


Figure C.1. Heaps and Zipf laws for the literary corpora varying *N*. The green curve corresponds to the the chosen value of N^*. The fitted values of β are in table C.6.

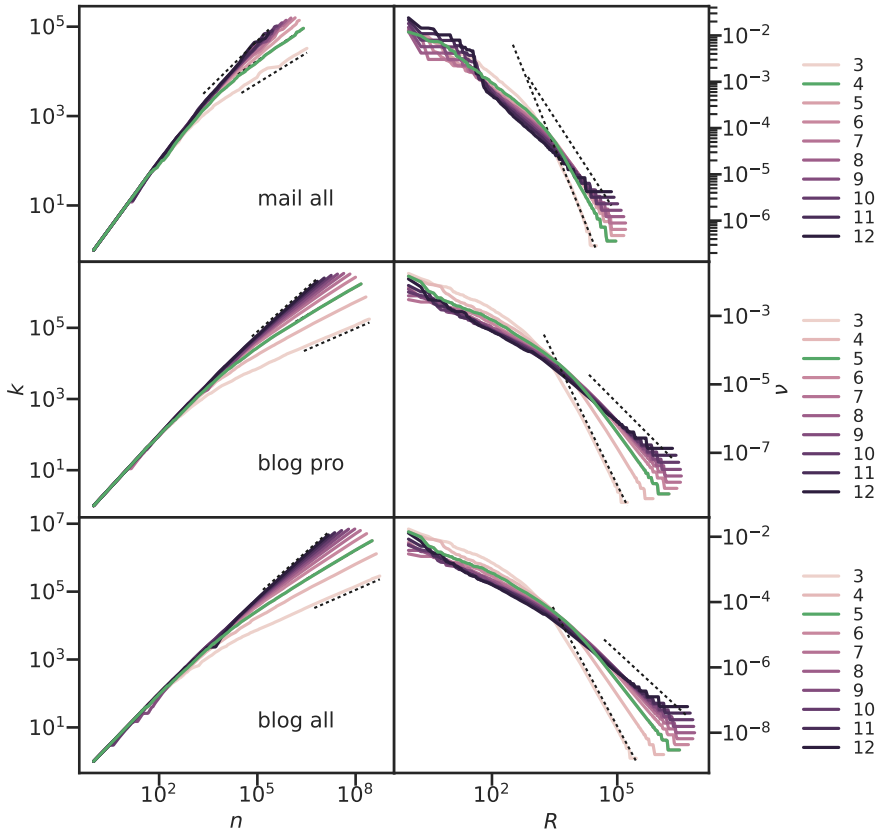


Figure C.2. Heaps and Zipf laws for the informal corpora varying *N*. The green curve corresponds to the the chosen value of N. The fitted values of β are in table C.6.

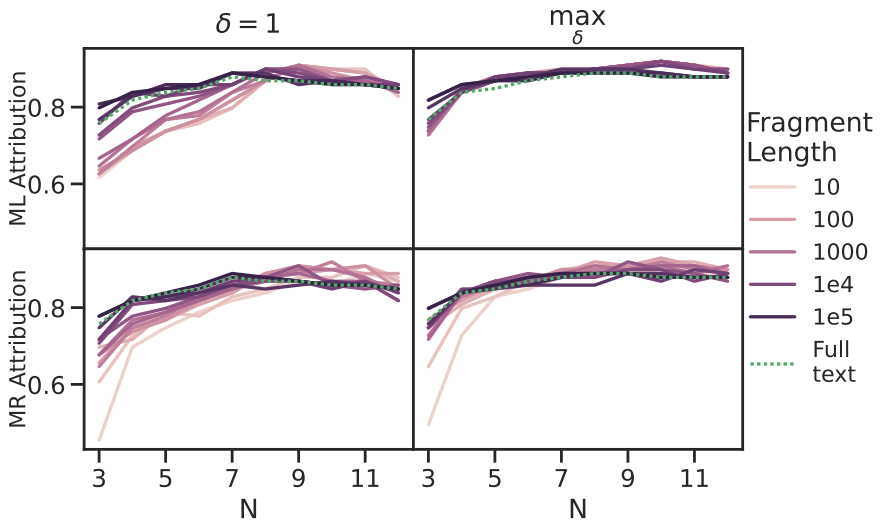
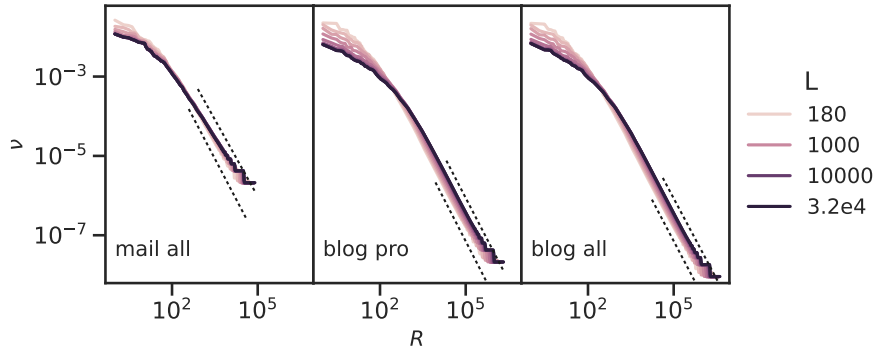
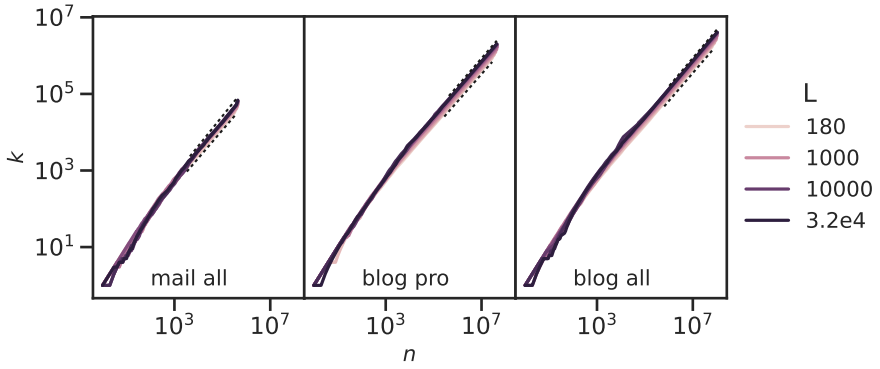


Figure C.3. Text attribution in the literary Polish corpus varying *N*. The maximum attribution scores with $\delta = 1$ are 90.9% using ML and 91.9% using MR. The use tuning of δ allows to attribute one book more, +1% with both ML and MR.



(a) Zipf's law



(b) Heaps' law

Figure C.4. Zipf's law and Heaps' law for the three informal corpora using *LZ77 sequences*. Straight lines in the Heaps' law plots show functions of the form $f(x) = ax^{\beta}$ for window lengths of 180 and 32000 characters. The exponent β equals to $\beta_{180} = 0.730$ and $\beta_{32000} = 0.774$ (Email), $\beta_{180} = 0.714$ and $\beta_{32000} = 0.721$ (Blog – prolific authors), $\beta_{180} = 0.730$ and $\beta_{32000} = 0.739$ (Blog – all authors). Straight lines in the Zipf's law plots show functions of the form $f(x) = ax^{-\alpha}$, where the exponent α is equal to β^{-1} for the different βs considered above. The differences between the different window lengths are less prominent than with the literary corpora.

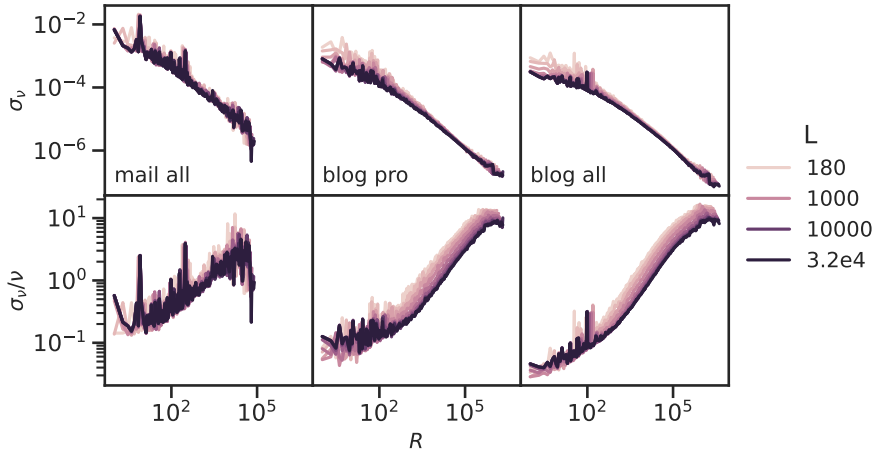


Figure C.5. Fluctuations in token frequency across authors using *LZ77 sequences* for the informal corpora. We show the standard deviation of token frequencies across authors. Tokens ordered by global frequency.

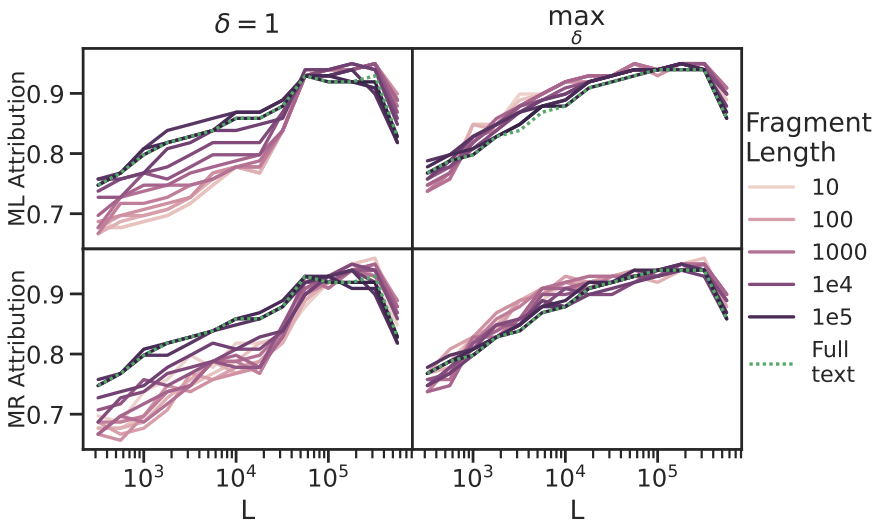


Figure C.6. Text attribution in the literary Polish corpus varying the window length. The maximum values are for window a length of 3.2×10^5 in all cases with a score of 94.9% using ML and 96.0% using MR. With both attribution techniques the maximum values tuning δ or not are the same.

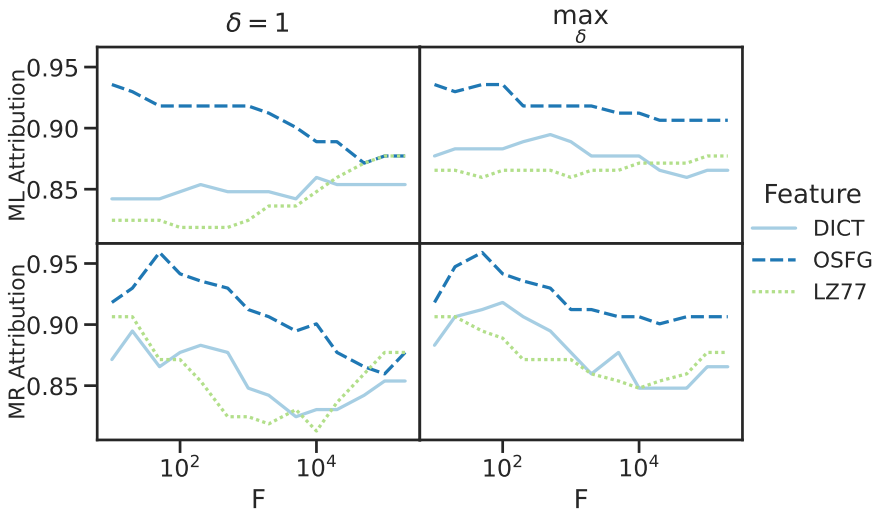


Figure C.7. Text attribution in the literary Italian corpus with the three different tokenisation techniques. The maximum values of attribution are 95.9% using MR and 93.6% using ML. The maximum values are independent from the tuning of δ.

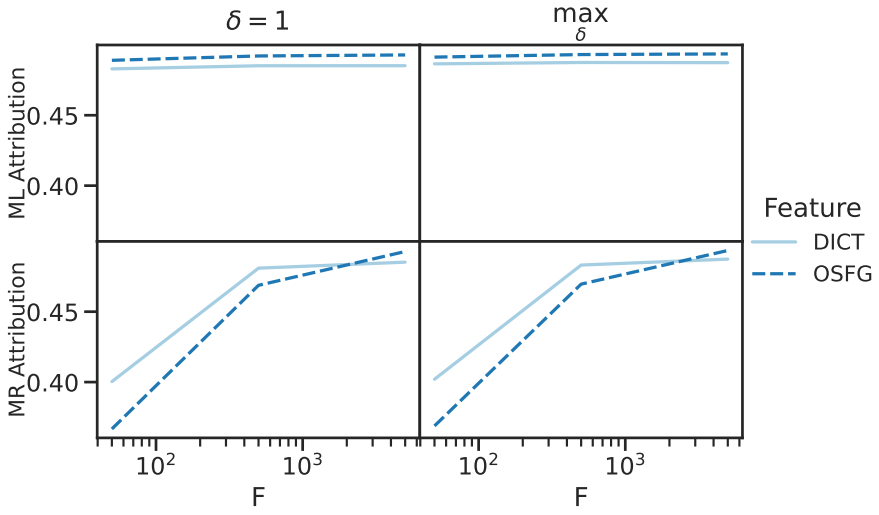


Figure C.8. Text attribution in the Blog corpus – prolific authors – with the two different tokenisation techniques. The maximum values of attribution are 49.28% with $\delta = 1$ and 49.35% Searching for the maximum over δ. The fraction of correct attribution between the ML and MR attribution differ by $\approx 10^{-5}$, few posts, in favour of ML.

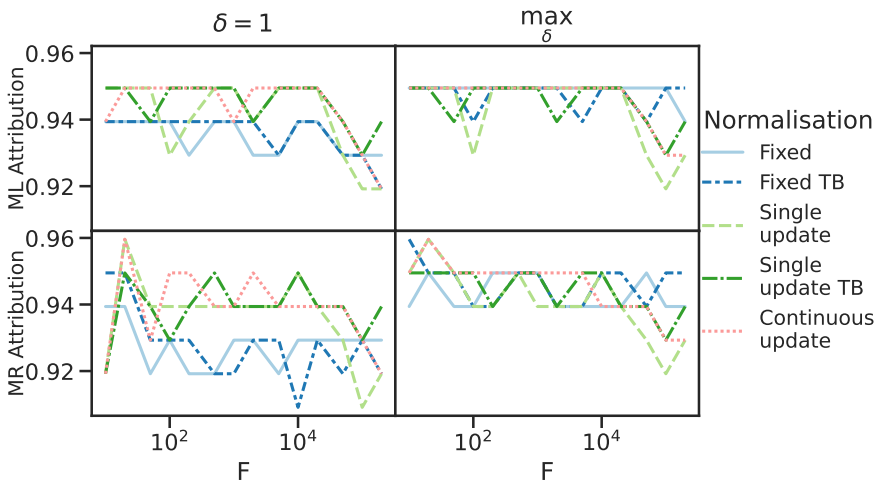


Figure C.9. Text attribution in the literary Polish corpus with the different choices of P_0. The normalisation curves marked as "TB" are Text Based, obtained counting the number of texts each token appears in. See section B.3 for a discussion of different token weighting. We report the best scores using the different choices of P_0 in table C.4.

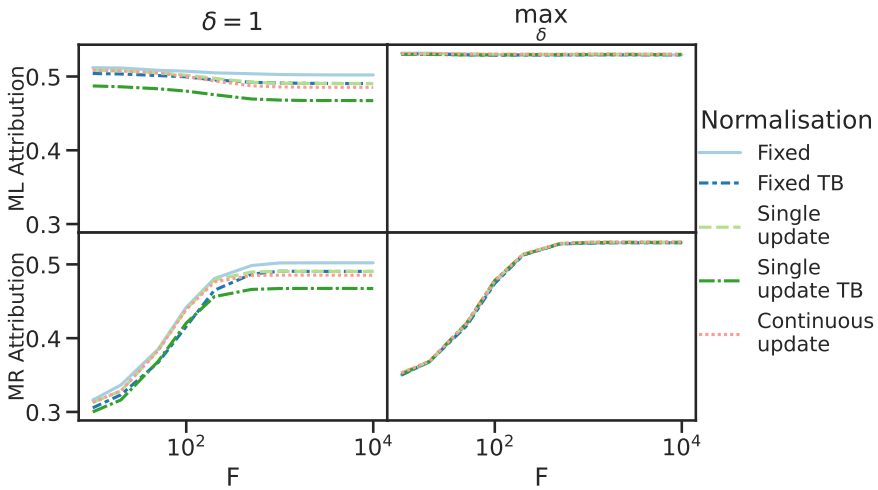


Figure C.10. Text attribution in the Email corpus with the different choices of P_0. The normalisation curves marked as "TB" are Text Based, obtained counting the number of texts each token appears in. See section B.3 for a discussion of different token weighting. We report the best scores using the different choices of P_0 in table C.5.

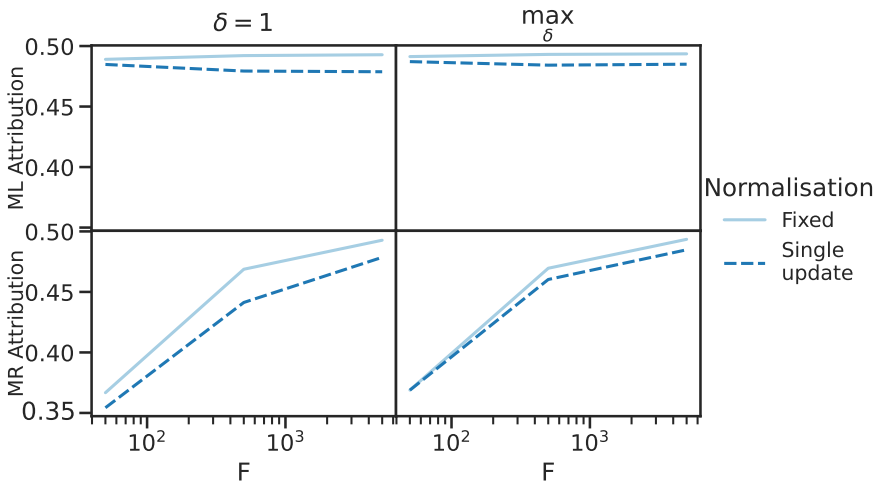


Figure C.11. Text attribution in the Blog corpus – prolific authors – with the different choices of P_0. Given the results obtained with the other corpora and the huge number of texts we excluded the text based P_0 from the analysis. As shown in fig. B.3 for the Blog corpus with all authors, we expect very little difference in the values of P_0 for the different tokens. We report the best scores using the different choices of P_0 in table C.5.

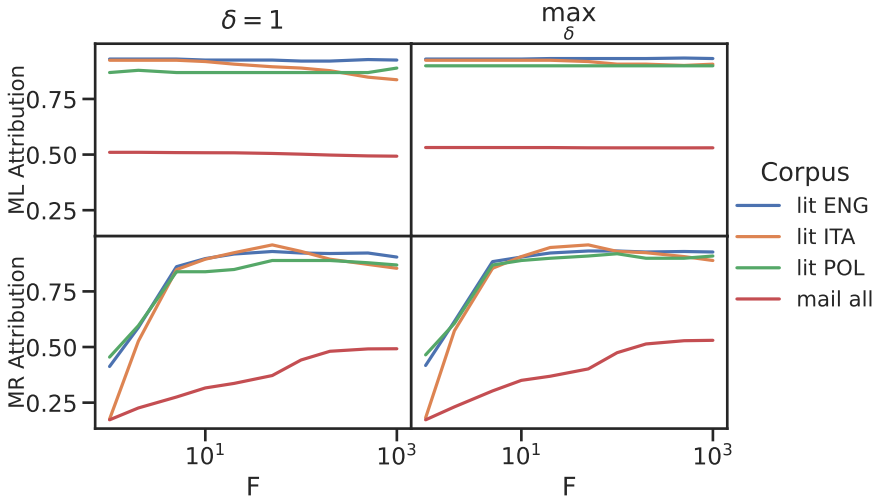


Figure C.12. Text attribution using *Overlapping Space Free N-grams* and short fragments. Fragment lengths in the range $[1, 1000]$ spaced one third of decade.

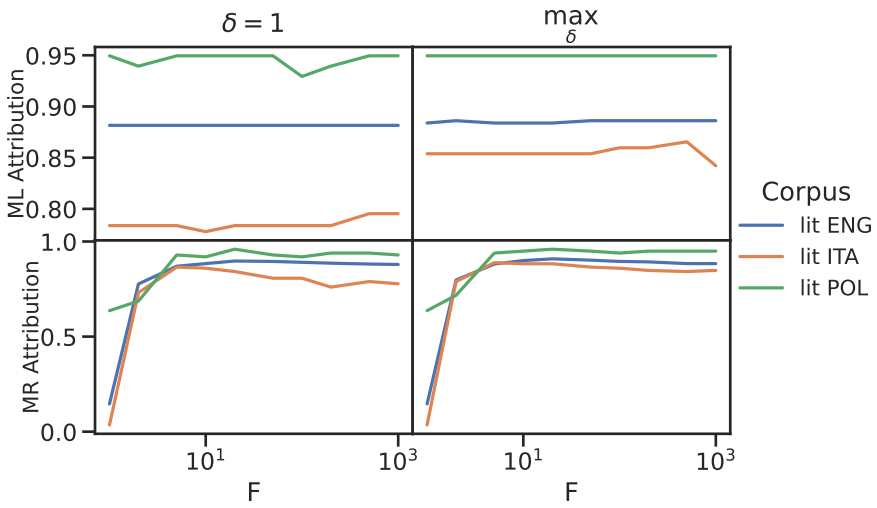


Figure C.13. Text attribution using *LZ77 sequences* and short fragments. Fragment lengths in the range $[1, 1000]$ spaced one third of decade.

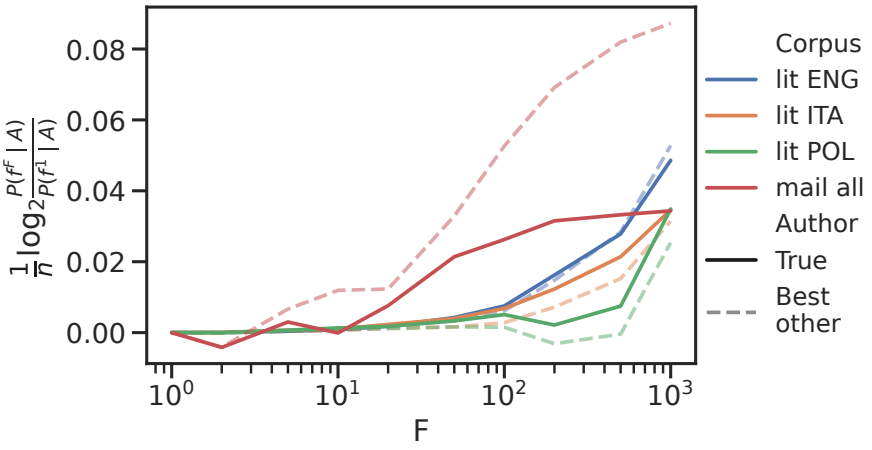


Figure C.14. Ratio of the average conditional probability per token varying fragment size over average probability using single token fragments. Results using *Dictionary words*. Due to the presence of outliers the average includes only the central 95% of the values.

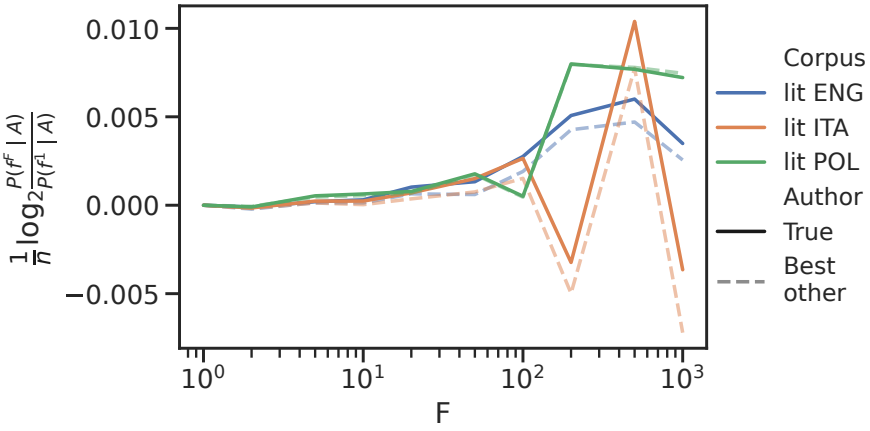


Figure C.15. Ratio of the average conditional probability per token varying fragment size over average probability using single token fragments. Results using *LZ77 sequences*. Due to the presence of outliers the average includes only the central 95% of the values.

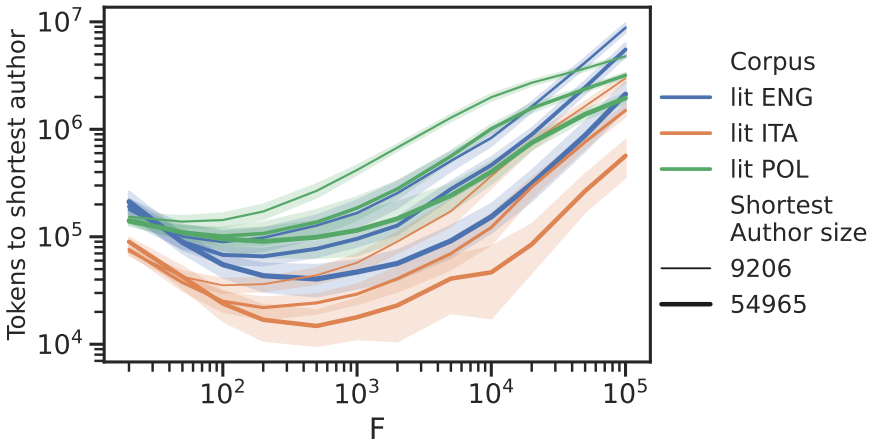


Figure C.16. Averages of the number of tokens in fragment attributed to the shortest author using *Dictionary words*. The weight of the lines is proportional to the length of the shortest author.

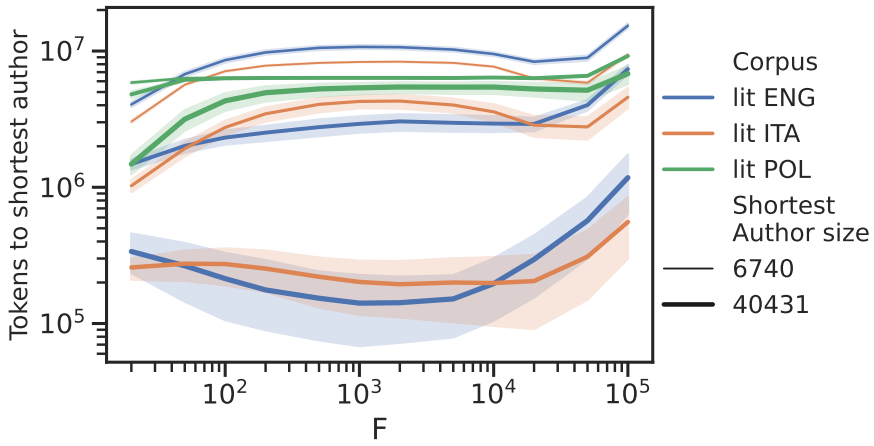


Figure C.17. Averages of the number of tokens in fragment attributed to the shortest author using *LZ77 sequences*. The weight of the lines is proportional to the length of the shortest author.

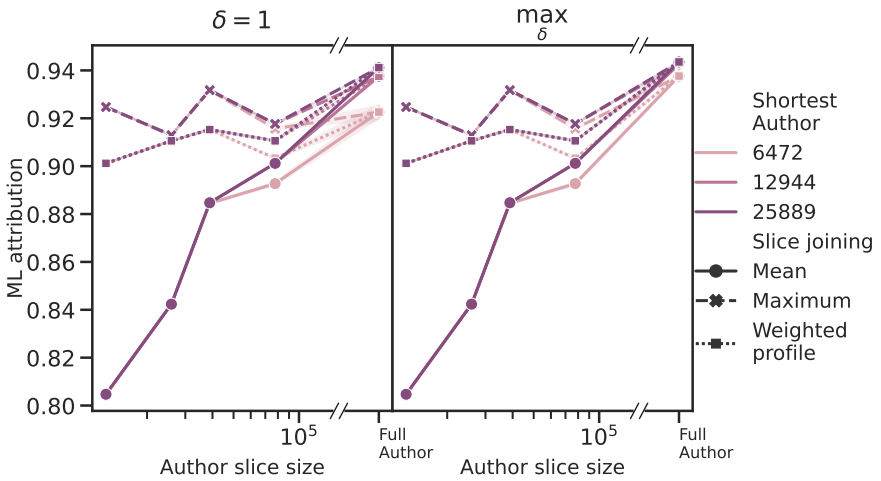


Figure C.18. Attribution in the literary English corpus varying the size of the author slices using *OSF N-grams*. The fragment length is fixed at $F = 10^4$. I do not report the results using MR attribution for conciseness as very close to those using Maximum Likelihood.

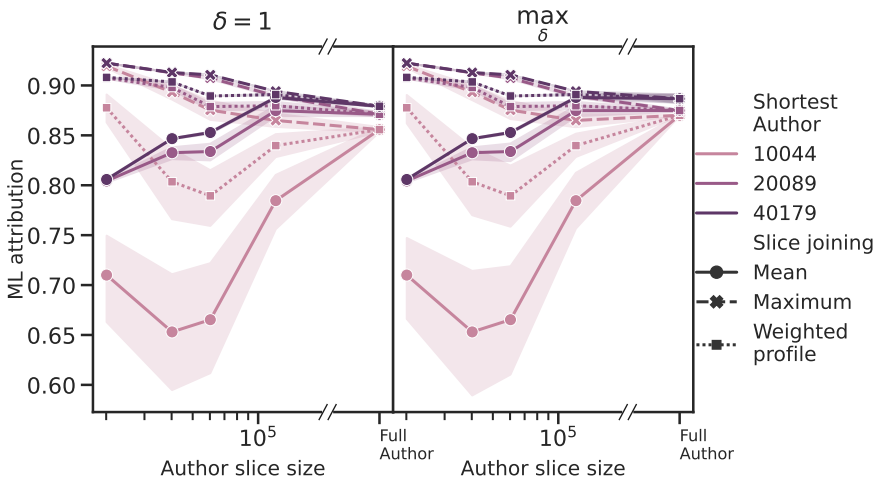


Figure C.19. Attribution in the literary English corpus varying the size of the author slices using *LZ77 sequences*. The fragment length is fixed at $F = 10^4$. I do not report the results using MR attribution for conciseness as very close to those using Maximum Likelihood.

Bibliography

[1] Abbasi, A. and Chen, H. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, **20** (2005), 67.

[2] Adams, D. *The Hitchhiker's Guide to the Galaxy*, vol. 1 of *The Hitchhiker's Guide to the Galaxy*. Pan Books (1979). ISBN 0-330-25864-8.

[3] Afroz, S., Brennan, M., and Greenstadt, R. Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE Symposium on Security and Privacy*, pp. 461–475. IEEE (2012).

[4] Afroz, S., Islam, A. C., Stolerman, A., Greenstadt, R., and McCoy, D. Doppelgänger finder: Taking stylometry to the underground. In *2014 IEEE Symposium on Security and Privacy*, pp. 212–226. IEEE (2014).

[5] Alias-i. LingPipe 4.1.2. http://http://www.alias-i.com/lingpipe/ (2011). Accessed August 2021.

[6] Allén, S. *Text Processing: Text Analysis and Generation, Text Typology and Attribution: Proceedings of Nobel Symposium 51*. 16. Coronet Books (1982).

[7] Amancio, D. R. Authorship recognition via fluctuation analysis of network topology and word intermittency. *Journal of Statistical Mechanics: Theory and Experiment*, **2015** (2015), P03005.

[8] Andersen, R. M. and May, R. M. Epidemiological parameters of HI V transmission. *Nature*, **333** (1988), 514.

[9] Antiqueira, L., Pardo, T. A. S., Nunes, M. d. G. V., and Oliveira Jr, O. N. Some issues on complex networks for author characterization. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, (2007), 51.

[10] Argamon, S. Interpreting Burrows's Delta: Geometric and probabilistic foundations. *Literary and Linguistic Computing*, **23** (2008), 131.

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

[12] Arun, R., Suresh, V., and Madhavan, C. V. Stopword graphs and authorship attribution in text corpora. In *2009 IEEE international conference on semantic computing*, pp. 192–196. IEEE (2009).

[13] Bagnall, D. Author Identification using multi-headed Recurrent Neural Networks—Notebook for PAN at CLEF 2015. In Cappellato et al. [29]. Available from: http://ceur-ws.org/Vol-1391.

[14] Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, **286** (1999), 509.

[15] Baronchelli, A., Caglioti, E., and Loreto, V. Artificial sequences and complexity measures. *Journal of Statistical Mechanics: Theory and Experiment*, **2005** (2005), P04002.

[16] Benedetto, D., Caglioti, E., and Loreto, V. Language Trees and Zipping. *Phys. Rev. Lett.*, **88** (2002), 048702. Available from: https://link.aps.org/doi/10.1103/PhysRevLett.88.048702, doi:10.1103/PhysRevLett.88.048702.

[17] Bessi, A. and Ferrara, E. Social bots distort the 2016 US Presidential election online discussion. *First monday*, **21** (2016).

[18] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *the Journal of machine Learning research*, **3** (2003), 993.

[19] Boyd, R. L. and Pennebaker, J. W. Did Shakespeare write Double Falsehood? Identifying individuals by creating psychological signatures with text analysis. *Psychological science*, **26** (2015), 570.

[20] Brainerd, B. The chronology of Shakespeare's plays: a statistical study. *Computers and the Humanities*, **14** (1980), 221. Available from: http://www.jstor.org/stable/30207349.

[21] Brennan, M., Afroz, S., and Greenstadt, R. Adversarial Stylometry: Circumventing Authorship Recognition to Preserve Privacy and Anonymity. *ACM Trans. Inf. Syst. Secur.*, **15** (2012). Available from: https://doi.org/10.1145/2382448.2382450, doi:10.1145/2382448.2382450.

[22] Buntine, W. Estimating likelihoods for topic models. In *Asian Conference on Machine Learning*, pp. 51–64. Springer (2009).

[23] Buntine, W. and Hutter, M. A Bayesian view of the Poisson-Dirichlet Process. Tech. Rep. arXiv:1007.0296, NICTA and ANU, Australia (2010). Available from: http://arxiv.org/abs/1007.0296.

[24] Burrows, J. 'Delta': a Measure of Stylistic Difference and a Guide to Likely Authorship. *Literary and Linguistic Computing*, **17** (2002), 267. Available from: https://doi.org/10.1093/llc/17.3.267, arXiv:https://academic.oup.com/dsh/article-pdf/17/3/267/2743069/170267.pdf, doi:10.1093/llc/17.3.267.

[25] Caliskan, A., Yamaguchi, F., Dauber, E., Harang, R., Rieck, K., Greenstadt, R., and Narayanan, A. When Coding Style Survives Compilation: De-anonymizing Programmers from Executable Binaries. *CoRR*, (2015). arXiv:1512.08546.

[26] Campbell, L. et al. *The SOPHISTES and POLITICUS of Plato*. Clarendon Press (1867).

[27] Can, F. and Patton, J. M. Change of writing style with time. *Computers and the Humanities*, **38** (2004), 61.

[28] Can, F. and Patton, J. M. Change of Word Characteristics in 20th-Century Turkish Literature: A Statistical Analysis. *Journal of Quantitative Linguistics*, **17** (2010), 167. Available from: https://doi.org/10.1080/09296174.2010.485444, arXiv:https://doi.org/10.1080/09296174.2010.485444, doi:10.1080/09296174.2010.485444.

[29] Cappellato, L., Ferro, N., Jones, G., and San Juan, E. (eds.). *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France*, CEUR Workshop Proceedings. CEUR-WS.org (2015). Available from: http://ceur-ws.org/Vol-1391.

[30] Chen, X., Kwong, S., and Li, M. A Compression Algorithm for DNA Sequences and Its Applications in Genome Comparison. *Genome Informatics*, **10** (1999), 51. doi:10.11234/gi1990.10.51.

[31] Clark, J. H. and Hannon, C. J. A Classifier System for Author Recognition Using Synonym-Based Features. In *MICAI 2007: Advances in Artificial Intelligence* (edited by A. Gelbukh and Á. F. Kuri Morales),

pp. 839–849. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). ISBN 978-3-540-76631-5.

[32] Contributors to the LDA project. lda: Topic modeling with latent Dirichlet allocation. https://github.com/lda-project/lda (2020). Accessed August 2021.

[33] Corominas-Murtra, B., Hanel, R., and Thurner, S. Understanding scaling through history-dependent processes with collapsing sample space. *Proceedings of the National Academy of Sciences*, **112** (2015), 5348.

[34] Corominas-Murtra, B., Hanel, R., and Thurner, S. Sample space reducing cascading processes produce the full spectrum of scaling exponents. *Scientific reports*, **7** (2017), 11223.

[35] Day, S., Brown, J., Thomas, Z., Gregory, I., Bass, L., and Dozier, G. Adversarial Authorship, AuthorWebs, and Entropy-Based Evolutionary Clustering. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6 (2016). doi:10.1109/ICCCN.2016.7568489.

[36] Day, S., Williams, H., Shelton, J., and Dozier, G. Towards the development of a Cyber Analysis & Advisement Tool (CAAT) for mitigating de-anonymization attacks. In *MAICS: The Modern Artificial Intelligence and Cognitive Science Conference* (2016).

[37] De Blasi, P., Favaro, S., Lijoi, A., Mena, R. H., Prünster, I., and Ruggiero, M. Are Gibbs-type priors the most natural generalization of the Dirichlet process? *IEEE transactions on pattern analysis and machine intelligence*, **37** (2015), 212.

[38] Deutsch, L. P. GZIP file format specification version 4.3. https://tools.ietf.org/pdf/rfc1952.pdf (1996). Accessed August 2021.

[39] Diaz, R. and Pariguan, E. On hypergeometric functions and Pochhammer k-symbol. *arXiv preprint math/0405596*, (2004).

[40] Diederich, J., Kindermann, J., Leopold, E., and Paass, G. Authorship attribution with support vector machines. *Applied intelligence*, **19** (2003), 109.

[41] Dingledine, R. and Mathewson, N. Design of a blocking-resistant anonymity system (2006).

[42] Eder, M. Style-markers in authorship attribution: a cross-language study of the authorial fingerprint. *Studies in Polish Linguistics*, **6** (2011).

[43] Eder, M. Does size matter? Authorship attribution, small samples, big problem. *Digital Scholarship in the Humanities*, **30** (2015), 167.

[44] Eder, M. and Rybicki, J. Do birds of a feather really flock together, or how to choose training samples for authorship attribution. *Literary and Linguistic Computing*, **28** (2012), 229.

[45] Eder, M., Rybicki, J., and Kestemont, M. Stylometry with R: A Package for Computational Text Analysis. *The R Journal*, **8** (2016), 107. Available from: https://doi.org/10.32614/RJ-2016-007, doi:10.32614/RJ-2016-007.

[46] El Manar El Bouanani, S. and Kassou, I. Authorship analysis studies: A survey. *International Journal of Computer Applications*, **86** (2014).

[47] Emmery, C., Manjavacas, E., and Chrupała, G. Style obfuscation by invariance. *arXiv preprint arXiv:1805.07143*, (2018).

[48] Estoup, J.-B. *Gammes Sténographiques*. Institut Stenographique de France (1916).

[49] Faust, C., Dozier, G., Xu, J., and King, M. C. Adversarial authorship, interactive evolutionary hill-climbing, and author CAAT-III. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8. IEEE (2017).

[50] Ferraz de Arruda, H., Nascimento Silva, F., Queiroz Marinho, V., Amancio, D., and da Fontoura Costa, L. Representation of texts as complex networks: a mesoscopic approach. *Journal of Complex Networks*, **6** (2017), 125. Available from: https://doi.org/10.1093/comnet/cnx023, arXiv:https://academic.oup.com/comnet/article-pdf/6/1/125/23677005/cnx023.pdf, doi:10.1093/comnet/cnx023.

[51] Ferrer-i Cancho, R. and Elvevåg, B. Random texts do not exhibit the real Zipf's law-like rank distribution. *PLoS One*, **5** (2010).

[52] Forsyth, R. S. Stylochronometry with substrings, or: a poet young and old. *Literary and Linguistic Computing*, **14** (1999), 467.

[53] Frantzeskou, G., Stamatatos, E., Gritzalis, S., and Katsikas, S. Effective Identification of Source Code Authors Using Byte-Level Information. In *Proceedings of the 28th International Conference on Software Engineering*, ICSE '06, pp. 893–896. Association for Computing Machinery, New York, NY, USA (2006). ISBN 1595933751. Available from: https://doi.org/10.1145/1134285.1134445, doi:10.1145/1134285.1134445.

[54] Ge, Z. and Sun, Y. Domain specific author attribution based on feedforward neural network language models. *arXiv preprint arXiv:1602.07393*, (2016).

[55] Ge, Z. and Sun, Y. Domain specific author attribution based on feedforward neural network language models. *arXiv preprint arXiv:1602.07393*, (2016).

[56] Gerlach, M. and Altmann, E. G. Scaling laws and fluctuations in the statistics of word frequencies. *New Journal of Physics*, **16** (2014), 113010.

[57] Griffiths, T. L. and Steyvers, M. Finding scientific topics. *Proceedings of the National academy of Sciences*, **101** (2004), 5228.

[58] Gull, M., Zia, T., and Ilyas, M. Source code author attribution using author's programming style and code smells. *International Journal of Intelligent Systems and Applications*, **9** (2017), 27.

[59] Heaps, H. S. *Information retrieval, computational and theoretical aspects*. Academic Press (1978).

[60] Herdan, G. *Type-token mathematics*, vol. 4. Mouton (1960).

[61] Hirst, G. and Feiguina, O. Bigrams of Syntactic Labels for Authorship Discrimination of Short Texts. *Literary and Linguistic Computing*, **22** (2007), 405. Available from: https://doi.org/10.1093/llc/fqm023, arXiv:https://academic.oup.com/dsh/article-pdf/22/4/405/2785853/fqm023.pdf, doi:10.1093/llc/fqm023.

[62] Hoppe, F. M. Pólya-like urns and the Ewens' sampling formula. *Journal of Mathematical Biology*, **20** (1984), 91.

[63] Hutson, M. Artificial intelligence unmasks anonymous chess players. *Science*, **375** (2022), 129.

[64] Iqbal, F., Binsalleeh, H., Fung, B. C., and Debbabi, M. Mining writeprints from anonymous e-mails for forensic investigation. *digital investigation*, **7** (2010), 56. doi:10.1016/j.diin.2010.03.003.

[65] Jafariakinabad, F. and Hua, K. A. Unifying Lexical, Syntactic, and Structural Representations of Written Language for Authorship Attribution. *SN Computer Science*, **2** (2021), 1.

[66] Jaynes, J. T. A search for trends in the poetic style of WB Yeats. *ALLC Journal*, **1** (1980), 11.

[67] Juola, P. The time course of language change. *Computers and the Humanities*, **37** (2003), 77.

[68] Juola, P. Future trends in authorship attribution. In *IFIP International Conference on Digital Forensics*, pp. 119–132. Springer (2007).

[69] Juola, P. and Baayen, R. H. A controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*, **20** (2005), 59.

[70] Kacmarcik, G. and Gamon, M. Obfuscating document stylometry to preserve author anonymity. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 444–451 (2006).

[71] Karlgren, J. Helander: An Authorship Attribution Case (2003). Available from: https://jussikarlgren.wordpress.com/2003/01/01/helander-an-authorship-attribution-case/.

[72] Kauffman, S. A. *The origins of order: Self-organization and selection in evolution*. OUP USA (1993).

[73] Kendal, W. S. A probabilistic model for the variance to mean power law in ecology. *Ecological modelling*, **80** (1995), 293.

[74] Keogh, E., Lonardi, S., and Ratanamahatana, C. A. Towards Parameter-Free Data Mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pp. 206–215. Association for Computing Machinery, New York, NY, USA (2004). ISBN 1581138881. doi:10.1145/1014052.1014077.

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

[76] Kjell, B. Authorship attribution of text samples using neural networks and Bayesian classifiers. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1660–1664. IEEE (1994).

[77] Klaussner, C. and Vogel, C. Stylochronometry: Timeline prediction in stylometric analysis. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 91–106. Springer (2015).

[78] Klimt, B. and Yang, Y. The Enron Corpus: A New Dataset for Email Classification Research. In *Machine Learning: ECML 2004* (edited by J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi), pp. 217–226. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). ISBN 978-3-540-30115-8.

[79] Kneser, R. and Ney, H. Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, vol. 1, pp. 181–184. IEEE (1995).

[80] Koppel, M., Schler, J., and Argamon, S. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, **60** (2009), 9.

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

[82] Koppel, M., Schler, J., and Bonchek-Dokow, E. Measuring Differentiability: Unmasking Pseudonymous Authors. *Journal of Machine Learning Research*, **8** (2007), 1261.

[83] Koppel, M. and Winter, Y. Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, **65** (2014), 178. arXiv:https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.22954, doi:10.1002/asi.22954.

[84] Kourtis, I. and Stamatatos, E. Author Identification Using Semi-supervised Learning—Notebook for PAN at CLEF 2011. In *Notebook Papers of CLEF 2011 Labs and Workshops, 19-22 September, Amsterdam, Netherlands* (edited by V. Petras, P. Forner, and P. Clough). CEUR-WS.org (2011). ISBN 978-88-904810-1-7. Available from: http://ceur-ws.org/Vol-1177.

[85] Krogh, A., Brown, M., Mian, I., Sjölander, K., and Haussler, D. Hidden Markov Models in Computational Biology: Applications to Protein Modeling. *Journal of Molecular Biology*, **235** (1994), 1501. Available from: https://www.sciencedirect.com/science/article/pii/S0022283684711041, doi:10.1006/jmbi.1994.1104.

[86] Kukushkina, O. V., Polikarpov, A. A., and Khmelev, D. V. Using literal and grammatical statistics for authorship attribution. *Problems of Information Transmission*, **37** (2001), 172.

[87] Kusakci, A. O. Authorship attribution using committee machines with k-nearest neighbors rated voting. In *11th Symposium on Neural Network Applications in Electrical Engineering*, pp. 161–166 (2012). doi:10.1109/NEUREL.2012.6419997.

[88] Lahiri, S. Complexity of word collocation networks: A preliminary structural analysis. *arXiv preprint arXiv:1310.5111*, (2013).

[89] Lalli, M., Tria, F., and Loreto, V. Data-Compression Approach to Authorship Attribution. In *Drawing Elena Ferrante's Profile: Workshop Proceedings, Padova, 7 September 2017* (edited by A. Tuzzi and M. A. Cortelazzo). Padova UP (2018).

[90] Lijoi, A., Muliere, P., Prünster, I., and Taddei, F. Innovation, growth and aggregate volatility from a Bayesian nonparametric perspective. *Electronic Journal of Statistics*, **10** (2016), 2179.

[91] Liu, Q., Yang, Y., Chen, C., Bu, J., Zhang, Y., and Ye, X. RNACompress: Grammar-based compression and informational complexity measurement of RNA secondary structure. *BMC bioinformatics*, **9** (2008), 1.

[92] Lü, L., Zhang, Z.-K., and Zhou, T. Zipf's law leads to Heaps' law: Analyzing their relation in finite-size systems. *PloS one*, **5** (2010), e14139.

[93] Lutoslawski, W. Principes de stylométrie appliqués à la chronologie des œuvres de Platon. *Revue des études grecques*, **11** (1898), 61.

[94] Mahmood, A., Ahmad, F., Shafiq, Z., Srinivasan, P., and Zaffar, F. A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X. *Proc. Priv. Enhancing Technol.*, **2019** (2019), 54.

[95] Marinho, V. Q., de Arruda, H. F., Lima, T. S., Costa, L. F., and Amancio, D. R. On the "Calligraphy" of Books. *arXiv preprint arXiv:1705.10415*, (2017).

[96] Marton, Y., Wu, N., and Hellerstein, L. On Compression-Based Text Classification. In *Advances in Information Retrieval* (edited by D. E. Losada and J. M. Fernández-Luna), pp. 300–314. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). ISBN 978-3-540-31865-1. doi:10.1007/978-3-540-31865-1_22.

[97] Mazzolini, A., Colliva, A., Caselle, M., and Osella, M. Heaps' law, statistics of shared components, and temporal patterns from a sample-space-reducing process. *Physical Review E*, **98** (2018), 052139.

[98] Mazzolini, A., Gherardi, M., Caselle, M., Lagomarsino, M. C., and Osella, M. Statistics of shared components in complex component systems. *Physical Review X*, **8** (2018), 021023.

[99] McDonald, A. W., Afroz, S., Caliskan, A., Stolerman, A., and Greenstadt, R. Use fewer instances of the letter "i": Toward writing style anonymization. In *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 299–318. Springer (2012).

[100] McIlroy-Young, R., Wang, R., Sen, S., Kleinberg, J., and Anderson, A. Detecting Individual Decision-Making Style: Exploring Behavioral Stylometry in Chess. *Advances in Neural Information Processing Systems*, **34** (2021).

[101] Mehri, A., Darooneh, A. H., and Shariati, A. The complex networks approach for authorship attribution of books. *Physica A: Statistical Mechanics and its Applications*, (2012), 2429.

[102] Mendenhall, T. C. The characteristic curves of composition. *Science*, **9** (1887), 237.

[103] Miller, G. A. WordNet: a lexical database for English. *Communications of the ACM*, **38** (1995), 39.

[104] Monechi, B., Ruiz-Serrano, Ã., Tria, F., and Loreto, V. Waves of novelties in the expansion into the adjacent possible. *PloS one*, **12** (2017), e0179303.

[105] Morozov, N. Linguistic Spectra: a means for distinguishing of plagiarism and original works for famous authors. A stylometry etude. Petrograd: Type of Imp. Acad. *Izvestiya otdeleniya russkogo yazy'ka i slovesnosti Imperatorskoj akademii nauk*, (1916), 1.

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

[107] Narayanan, A., Paskov, H., Gong, N. Z., Bethencourt, J., Stefanov, E., Shin, E. C. R., and Song, D. On the feasibility

of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*, pp. 300–314. IEEE (2012).

[108] Neal, T., Sundararajan, K., Fatima, A., Yan, Y., Xiang, Y., and Woodard, D. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSUR)*, **50** (2017), 1.

[109] Nirmal.A, N., Sohn, K.-A., Chung, T.-S., et al. A graph model based author attribution technique for single-class e-mail classification. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, pp. 191–196. IEEE (2015). doi:10.1109/ICIS.2015.7166592.

[110] Oliveira, W., Justino, E., and Oliveira, L. Comparing compression models for authorship attribution. *Forensic Science International*, **228** (2013), 100. Available from: https://www.sciencedirect.com/science/article/pii/S0379073813000923, doi:10.1016/j.forsciint.2013.02.025.

[111] Patchala, J. and Bhatnagar, R. Authorship attribution by consensus among multiple features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2766–2777 (2018).

[112] Peersman, C., Daelemans, W., and Van Vaerenbergh, L. Predicting Age and Gender in Online Social Networks. In *Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents*, SMUC '11, pp. 37–44. Association for Computing Machinery, New York, NY, USA (2011). ISBN 9781450309493. Available from: https://doi.org/10.1145/2065023.2065035, doi:10.1145/2065023.2065035.

[113] Peng, F., Schuurmans, D., and Wang, S. Augmenting naive bayes classifiers with statistical language models. *Information Retrieval*, **7** (2004), 317.

[114] Pitman, J. *Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII-2002*. Springer (2006).

[115] Pritchard, J. K., Stephens, M., and Donnelly, P. Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, **155** (2000), 945. Available from: https://www.genetics.org/content/155/2/945, arXiv:https://www.genetics.org/content/155/2/945.full.pdf.

[116] Raghunadha Reddy, T., Lakshminarayana, M., Vishnu Vardhan, B., Sai Prasad, K., and Amarnath Reddy, E. A New Document Representation Approach for Gender Prediction Using Author Profiles. In *First International Conference on Artificial Intelligence and Cognitive Computing* (edited by R. S. Bapi, K. S. Rao, and M. V. N. K. Prasad), pp. 39–47. Springer Singapore, Singapore (2019). ISBN 978-981-13-1580-0.

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

[118] Rangel, F., Montes-y-Gómez, M., Potthast, M., and Stein, B. Overview of the 6th Author Profiling Task at PAN 2018: Cross-domain Authorship Attribution and Style Change Detection. In *CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France* (edited by L. Cappellato, N. Ferro, J.-Y. Nie, and L. Soulier), CEUR Workshop Proceedings. CEUR-WS.org (2018). Available from: http://ceur-ws.org/Vol-2125/.

[119] Rangel, F. and Rosso, P. Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In *CLEF 2019 Labs and Workshops, Notebook Papers* (edited by L. Cappellato, N. Ferro, D. Losada, and H. Müller), CEUR Workshop Proceedings. CEUR-WS.org (2019). Available from: http://ceur-ws.org/Vol-2380/.

[120] Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., and Inches, G. Overview of the Author Profiling Task at PAN 2013. In *CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain* (edited by P. Forner, R. Navigli, and D. Tufis). CEUR-WS.org (2013). ISBN 978-88-904810-3-1. Available from: http://ceur-ws.org/Vol-1179.

[121] Rao, J. R., Rohatgi, P., et al. Can pseudonymity really guarantee privacy? In *USENIX Security Symposium*, pp. 85–96 (2000).

[122] Reddy, T. R., Vardhan, B. V., and Reddy, P. V. A survey on authorship profiling techniques. *International Journal of Applied Engineering Research*, **11** (2016), 3092.

[123] Romanov, A., Kurtukova, A., Fedotova, A., and Meshcheryakov, R. Natural Text Anonymization Using Universal Transformer

with a Self-attention. In *Proceedings of the III International Conference on Language Engineering and Applied Linguistics (PRLEAL-2019), Saint Petersburg, Russia*, pp. 22–37 (2019).

[124] Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, **20** (1987), 53. Available from: https://www.sciencedirect.com/science/article/pii/0377042787901257, doi:10.1016/0377-0427(87)90125-7.

[125] Rybicki, J. and Eder, M. Deeper Delta across genres and languages: do we really need the most frequent words? *Literary and Linguistic Computing*, **26** (2011), 315. Available from: https://doi.org/10.1093/llc/fqr031, arXiv:https://academic.oup.com/dsh/article-pdf/26/3/315/3955977/fqr031.pdf, doi:10.1093/llc/fqr031.

[126] Saedi, C. and Dras, M. Siamese networks for large-scale author identification. *Computer Speech & Language*, **70** (2021), 101241. Available from: https://www.sciencedirect.com/science/article/pii/S0885230821000486, doi:10.1016/j.csl.2021.101241.

[127] Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjölander, K., Underwood, R. C., and Haussler, D. Stochastic context-free grammers for tRNA modeling. *Nucleic Acids Research*, **22** (1994), 5112. Available from: https://doi.org/10.1093/nar/22.23.5112, arXiv:https://academic.oup.com/nar/article-pdf/22/23/5112/7122783/22-23-5112.pdf, doi:10.1093/nar/22.23.5112.

[128] Sanderson, C. and Guenter, S. Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 482–491 (2006).

[129] Savoy, J. Authorship attribution based on a probabilistic topic model. *Information Processing & Management*, **49** (2013), 341.

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

[131] Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. W. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, vol. 6, pp. 199–205 (2006).

[132] Searls, D. B. The language of genes. *Nature*, **420** (2002), 211.

[133] Seroussi, Y., Bohnert, F., and Zukerman, I. Authorship attribution with author-aware topic models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 264–269 (2012).

[134] Seroussi, Y., Zukerman, I., and Bohnert, F. Authorship attribution with latent Dirichlet allocation. In *Proceedings of the fifteenth conference on computational natural language learning*, pp. 181–189 (2011).

[135] Seroussi, Y., Zukerman, I., and Bohnert, F. Authorship attribution with topic models. *Computational Linguistics*, **40** (2014), 269.

[136] Shetty, R., Schiele, B., and Fritz, M. A4NT: author attribute anonymity by adversarial training of neural machine translation. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 1633–1650 (2018).

[137] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., and Chanona-Hernández, L. Syntactic N-grams as machine learning features for natural language processing. *Expert Systems with Applications*, **41** (2014), 853. Methods and Applications of Artificial and Computational Intelligence. Available from: https://www.sciencedirect.com/science/article/pii/S0957417413006271, doi:10.1016/j.eswa.2013.08.015.

[138] Simon, H. A. On a class of skew distribution functions. *Biometrika*, **42** (1955), 425.

[139] Stamatatos, E. Authorship attribution based on feature set subspacing ensembles. *International Journal on Artificial Intelligence Tools*, **15** (2006), 823.

[140] Stamatatos, E. Author Identification Using Imbalanced and Limited Training Texts. In *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*, pp. 237–241 (2007). doi:10.1109/DEXA.2007.5.

[141] Stamatatos, E. Author identification: Using text sampling to handle the class imbalance problem. *Information Processing & Management*, **44** (2008), 790.

[142] Stamatatos, E. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, **60** (2009), 538.

[143] Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López López, A., Potthast, M., and Stein, B. Overview of the Author Identification Task at PAN 2015. In Cappellato et al. [29]. Available from: http://ceur-ws.org/Vol-1391/.

[144] Stamou, C. Stylochronometry: Stylistic development, sequence of composition, and relative dating. *Literary and Linguistic Computing*, **23** (2007), 181.

[145] Stella, M., Ferrara, E., and De Domenico, M. Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences*, **115** (2018), 12435. Available from: https://www.pnas.org/content/115/49/12435, arXiv:https://www.pnas.org/content/115/49/12435.full.pdf, doi:10.1073/pnas.1803470115.

[146] Steyvers, M. and Griffiths, T. Probabilistic topic models. In *Handbook of latent semantic analysis*, pp. 439–460. Psychology Press (2007).

[147] Taylor, L. R. Aggregation, variance and the mean. *Nature*, **189** (1961), 732.

[148] Teh, Y. W. A Bayesian interpretation of interpolated Kneser-Ney. *TRA2/06*, (2006).

[149] Tria, F., Loreto, V., and Servedio, V. Zipf's, Heaps' and Taylor's Laws are Determined by the Expansion into the Adjacent Possible. *Entropy*, **20** (2018), 752.

[150] Tria, F., Loreto, V., Servedio, V., and Strogatz, S. The dynamics of correlated novelties. *Scientific reports*, **4** (2014), 1.

[151] Valla, L. Discourse on the Forgery of the Alleged Donation of Constantine. *Latin and English. Trans. Christopher B. Coleman. New Haven*, (1922).

[152] Verhoeven, B. and Daelemans, W. CLiPS Stylometry Investigation (CSI) corpus: A Dutch corpus for the detection of age, gender, personality, sentiment and deception in text. In *LREC*, pp. 3081–3085 (2014).

[153] Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. Evaluation Methods for Topic Models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1105–1112. Association for Computing Machinery, New York, NY, USA (2009). ISBN 9781605585161. Available from: https://doi.org/10.1145/1553374.1553515, doi:10.1145/1553374.1553515.

[154] Whissell, C. Traditional and emotional stylometric analysis of the songs of Beatles Paul McCartney and John Lennon. *Computers and the Humanities*, **30** (1996), 257.

[155] Wyner, A. D. and Ziv, J. The sliding-window Lempel-Ziv algorithm is asymptotically optimal. *Proceedings of the IEEE*, **82** (1994), 872.

[156] Yang, M., Mei, J., Xu, F., Tu, W., and Lu, Z. Discovering author interest evolution in topic modeling. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 801–804 (2016).

[157] Yang, M., Zhu, D., Tang, Y., and Wang, J. Authorship attribution with topic drift model. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[158] Yule, G. U. *The statistical study of literary vocabulary.* Cambridge University Press (1944).

[159] Zanette, D. and Montemurro, M. Dynamics of text generation with realistic Zipf's distribution. *Journal of quantitative Linguistics*, **12** (2005), 29.

[160] Zangerle, E., Mayerl, M., Potthast, M., and Stein, B. Overview of the Style Change Detection Task at PAN 2021. In *CLEF 2021 Labs and Workshops, Notebook Papers* (edited by G. Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi), CEUR Workshop Proceedings. CEUR-WS.org (2021). Available from: http://ceur-ws.org/Vol-2936/.

[161] Zhao, Y. and Zobel, J. Entropy-based authorship search in large document collections. In *European Conference on Information Retrieval*, pp. 381–392. Springer (2007).

[162] Zheng, R., Li, J., Chen, H., and Huang, Z. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and*

Technology, **57** (2006), 378. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20316, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.20316, doi:10.1002/asi.20316.

[163] Zipf, G. K. *The psycho-biology of language: An introduction to dynamic philology*. Routledge (1936).

[164] Ziv, J. and Lempel, A. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, **23** (1977), 337.