Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Artificial sequences and complexity measures

Andrea Baronchelli[1,2], **Emanuele Caglioti**[3] **and Vittorio Loreto**[1,2]

[1] Physics Department, 'La Sapienza' University, Piazzale Aldo Moro 5, 00185 Rome, Italy
[2] INFM-SMC, Unità di Roma 1, Italy
[3] Mathematics Department, 'La Sapienza' University, Piazzale Aldo Moro 5, 00185 Rome, Italy
E-mail: baronka@pil.phys.uniroma1.it, caglioti@mat.uniroma1.it and loreto@roma1.infn.it

Received 4 November 2004
Accepted 21 March 2005
Published 5 April 2005

Online at stacks.iop.org/JSTAT/2005/P04002
doi:10.1088/1742-5468/2005/04/P04002

Abstract. In this paper we exploit concepts of information theory to address the fundamental problem of identifying and defining the most suitable tools for extracting, in a automatic and agnostic way, information from a generic string of characters. We introduce in particular a class of methods which use in a crucial way data compression techniques in order to define a measure of remoteness and distance between pairs of sequences of characters (e.g. texts) based on their relative information content. We also discuss in detail how specific features of data compression techniques could be used to introduce the notion of *dictionary* of a given sequence and of *artificial text* and we show how these new tools can be used for information extraction purposes. We point out the versatility and generality of our method that applies to any kind of corpora of character strings independently of the type of coding behind them. We consider as a case study linguistic motivated problems and we present results for automatic language recognition, authorship attribution and self-consistent classification.

Keywords: analysis of algorithms, source and channel coding, new applications of statistical mechanics

1742-5468/05/P04002+26$30.00

Contents

1. Introduction

One of the most challenging issues of recent years is presented by the overwhelming mass of available data. While this abundance of information and the extreme accessibility of it represents an important cultural advance, it raises on the other hand the problem of retrieving relevant information. Imagine entering the largest library in the world, seeking all relevant documents on your favourite topic. Without the help of an efficient librarian this would be a difficult, perhaps hopeless, task. The desired references would probably remain buried under tons of irrelevancies. Clearly the need for effective tools for information retrieval and analysis is becoming more urgent as the databases continue to grow.

First of all let us consider some among the possible sources of information. In Nature many systems and phenomena are often represented in terms of sequences or strings of characters. In experimental investigations of physical processes, for instance, one typically has access to the system only through a measuring device which produces a time record of a certain observable, i.e. a sequence of data. On the other hand other systems are intrinsically described by strings of characters, e.g. DNA and protein sequences, language.

When analysing a string of characters the main aim is to extract the information it provides. For a DNA sequence this would correspond, for instance, to the identification of the subsequences codifying the genes and their specific functions. On the other hand for a written text one is interested in questions like recognizing the language in which the text is written, its author or the subject treated.

One of the main approaches to this problem, the one we address in this paper, is that of information theory (IT) [1, 2] and in particular the theory of data compression.

J. Stat. Mech. (2005) P04002

In a recent letter [3] a method for context recognition and context classification of strings of characters or other equivalent coded information has been proposed. The remoteness between two sequences A and B was estimated by zipping a sequence $A+B$ obtained by appending the sequence B after the sequence A and exploiting the features of data compression schemes like *gzip* (whose core is provided by the Lempel–Ziv 77 (LZ77) algorithm [4]). This idea was used for authorship attribution and, by defining a suitable distance between sequences, for languages phylogenesis.

The idea of appending two files and zipping the resulting file in order to measure the remoteness between them had been previously proposed by Loewenstern *et al* [5] (using *zdiff* routines) who applied it to the analysis of DNA sequences, and by Khmelev [6] who applied the method to authorship attribution. Similar methods have been proposed by Juola [7], Teahan [8] and Thaper [9].

In this paper we extend the analysis of [3] and we describe in detail the methods used to define and measure the remoteness (or similarity) between pairs of sequences based on their relative informatic content. We devise in particular, without loss of generality with respect to the nature of the strings of characters, a method for measuring this *distance* based on data compression techniques.

The principal tool for the application of these methods is the LZ77 algorithm, which, roughly speaking, achieves the compression of a file exploiting the presence of repeated subsequences. We introduce (see also [10]) the notion of the *dictionary* of a sequence, defined as the set of all the repeated substrings found by LZ77 in a sequential parsing of a file, and we refer to these substrings as the dictionary's *words*. Besides being of great intrinsic interest, every dictionary allows for the creation of *artificial texts* (AT) obtained by the concatenation of random extracted words. In this paper we discuss how comparing AT, instead of the original sequences, could represent a valuable and coherent tool for information extraction to be used in very different domains. We then propose a general AT comparison scheme (ATC) and show that it yields to remarkable results in experiments.

We have chosen for our tests some textual corpora and we have evaluated our method on the basis of the results obtained on some linguistic motivated problems. Is it possible to automatically recognize the language in which a given text is written? Is it possible to automatically guess the author and the subject of a given text? And finally is it possible to define methods for the automatic classification of the texts of a given corpus?

The choice of the linguistic framework is justified by the fact that this is a field where anybody could be able to judge, at least partially, the validity and the relevance of the results. Since we are introducing techniques for which a benchmark does not exist it is important to check their validity with known and controlled examples. This does not mean that the range of applicability is reduced to linguistics. On the contrary the ambition is to provide physicists with tools which could parallel other standard tools for analysing strings of characters.

In this perspective it is worthwhile recalling here some of the latest developments of sequence analysis in physics related problems. A first field of activity [11, 12] is that of segmentation problems, i.e. cases in which a unique string must be partitioned into subsequences according to some criteria to identify discontinuities in its statistical properties. A classical example is that of the separation of coding and non-coding portions in the DNA but the analysis of genetic sequences in general represents a very rich source of segmentation problems (see, for instance, [10], [13]–[15]).

J. Stat. Mech. (2005) P04002

A more recent area is represented by the use of data compression techniques to test specific properties of symbolic sequences. In [16], the technology behind adaptive dictionary data compression algorithms is used in a suitable way (which is very close to our approach) as an estimate of reversibility of time series, as well as a statistical likelihood test. Another interesting field is related to the problem of the generation of random numbers. In [17] the importance of suitable measures of conditional entropies, in order to check the real level of randomness of random numbers, is outlined and an entropic approach is used to discuss some random number generator shortcomings (see also [18]).

Finally, another area of interest is represented by the use of data compression techniques to estimate entropic quantities (e.g. Shannon entropy, algorithmic complexity, Kullback–Leibler divergence). Even though not new this area is still topical [19, 20]. A specific application that has generated an interesting debate concerns the analysis of electroencephalograms of epilepsy patients [21]–[23]. In particular in these papers it is argued that measures like the Kullback–Leibler divergence could be used to spot information in medical data. The debate is wide open.

The outline of the paper is as follows. In section 2, after a short theoretical introduction, we recall how data compression techniques could be used to evaluate entropic quantities. In particular we recall the definition of the LZ77 [4] compression algorithm and we address the problem of using it to evaluate quantities like the relative entropy of two generic sequences as well as to define a suitable distance between them. In section 3 we introduce the concept of artificial text (AT) and present a method for information extraction based on artificial text comparison. Sections 4 and 5 are devoted to the results obtained with our method in two different contexts: the recognition and extraction of linguistic features (section 4) and the self-consistent classification of large corpora (section 5). Finally section 6 is devoted to the conclusions and to a short discussion about possible perspectives.

2. Complexity measures and data compression

Before entering into the details of our method let us briefly recall the definition of entropy of a string. Shannon's definition of information entropy is indeed a probabilistic concept referring to the source emitting strings of characters.

Consider a symbolic sequence $(\sigma_1\,\sigma_2\,\cdots)$, where σ_t is the symbol emitted at time t and each σ_t can assume one of m different values. Assuming that the sequence is stationary we introduce the N-block entropy:

$$H_N = -\sum_{\{W_N\}} p(W_N)\ln p(W_N) \tag{1}$$

where $p(W_N)$ is the probability of the N-word $W_N = (\sigma_t\,\sigma_{t+1}\,\cdots\,\sigma_{t+N-1})$, and $\ln = \log_e$. The differential entropies

$$h_N = H_{N+1} - H_N \tag{2}$$

have a rather obvious meaning: h_N is the average information supplied by the $(N+1)$th symbol, provided the N previous ones are known. Noting that the knowledge of a longer past history cannot increase the uncertainty in the next outcome, one has that h_N cannot

J. Stat. Mech. (2005) P04002

increase with N, i.e. $h_{N+1} \leq h_N$. With these definitions the Shannon entropy for an ergodic stationary process is defined as

$$h = \lim_{N\to\infty} h_N = \lim_{N\to\infty} \frac{H_N}{N}. \tag{3}$$

It is easy to see that for a kth-order Markov process (i.e. one such that the conditional probability of having a given symbol only depends on the last k symbols, $p(\sigma_t|\sigma_{t-1}\,\sigma_{t-2},\ldots) = p(\sigma_t|\sigma_{t-1}\,\sigma_{t-2},\ldots,\sigma_{t-k})$), we have $h_N = h$ for $N \geq k$.

The Shannon entropy h measures the average amount of information per symbol and it is an estimate of the 'surprise' the source emitting the sequence reserves to us. It is remarkable that, under rather natural assumptions, the entropy H_N, apart from a multiplicative factor, is the unique quantity which characterizes the 'surprise' of the N-words [24]. Let us try to explain in which sense entropy can be considered as a measure of surprise. Suppose that the surprise one feels upon learning that an event E has occurred depends only on the probability of E. If the event occurs with probability 1 (sure) our surprise at its occurring will be zero. On the other hand if the probability of occurrence of the event E is quite small our surprise will be proportionally large. For a single event occurring with probability p the surprise is proportional to $\ln p$. Let us consider now a random variable X, which can take N possible values $x_1,\ldots,x_N$ with probabilities $p_1,\ldots,p_N$; the expected amount of surprise we shall experience upon learning the value of X is given precisely by the entropy of the source emitting the random variable X, i.e. $-\sum p_i \ln p_i$.

The definition of entropy is closely related to a very old problem, that of transmitting a message without losing information, i.e. the problem of efficient encoding [25].

A good example is the Morse code. In the Morse code a text is encoded with two characters: line and dot. What is the best way to encode the characters of the English language (provided one can define a source for English) with sequences of dots and lines? The idea of Morse was to encode the more frequent characters with the minimum number of characters. Therefore 'e' which is the most frequent English letter is encoded with one dot ($\cdot$), while the letter q is encoded with three lines and one dot ($--\cdot-$).

The problem of the optimal coding for a text (or an image or any other kind of information) has been enormously studied. In particular, Shannon [1] showed that there is a limit on the possibility of encoding a given sequence. This limit is the entropy of the sequence.

This result is particularly important when the aim is the measure of the information content of a single finite sequence, without any reference to the source that emitted it. In this case the reference framework is the algorithmic complexity theory and the basic concept is Chaitin–Kolmogorov entropy or algorithmic complexity (AC) [26]–[29]: *the entropy of a string of characters is the length (in bits) of the smallest program which produces as output the string and stops afterwards.* This definition is really abstract. In particular it is impossible, even in principle, to find such a program and as a consequence the algorithmic complexity is a non-computable quantity. This impossibility is related to the halting problem and to Godel's theorem [30].

It is important to recall that there exists a rather important relation between the algorithmic complexity $K_N(W_N)$ of a sequence W_N of N characters and H_N:

$$\frac{1}{N}\langle K_N\rangle = \frac{1}{N}\sum_{W_N} K_N(W_N)P(W_N) \xrightarrow[N\to\infty]{} \frac{h}{\ln 2} \tag{4}$$

where K_N is the binary length of the shorter program needed to specify the sequence W_N.

J. Stat. Mech. (2005) P04002

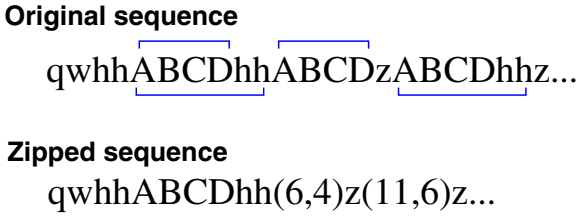

Figure 1. Scheme of the LZ77 algorithm: the LZ77 algorithm works sequentially and at a generic step looks in the look-ahead buffer for substrings already encountered in the buffer already scanned. These substrings are replaced by a pointer (d, n) where d is the distance of the previous occurrence of the same substring and n is its length. Only strings longer than two characters are replaced in the example.

As a consequence there exists a relation between the maximum compression rate of a sequence $(\sigma_1\,\sigma_2\,\cdots)$ expressed in an alphabet with m symbols, and h. If the length N of the sequence is large enough, then it is not possible to compress it into another sequence (with an alphabet with m symbols) whose size is smaller than $Nh/\ln m$. Therefore, noting that the number of bits needed for a symbol in an alphabet with m symbols is $\ln m$, one has that the maximum allowed compression rate is $h/\ln m$ [1].

Though the maximal theoretical limit of the algorithmic complexity is not achievable, there are nevertheless algorithms explicitly conceived to approach it. These are the file compressors or zippers. A zipper takes a file and tries to transform it into the shortest possible file. Obviously this is not the best way to encode the file but it represents a good approximation to it.

A great improvement in the field of data compression has been represented by the Lempel and Ziv algorithm (LZ77) [4] (used for instance by *gzip* and *zip*). It is interesting to briefly recall how it works (see figure 1). Let $x = x_1, \ldots, x_N$ be the sequence to be zipped, where x_i represents a generic character of a sequence's alphabet. The LZ77 algorithm finds duplicated strings in the input data. The second occurrence of a string is replaced by a pointer to the previous string given by two numbers: a distance, representing how far back into the window the sequence starts, and a length, representing the number of characters for which the sequence is identical. More specifically the algorithm proceeds sequentially along the sequence. Let us suppose that the first n characters have been codified. Then the zipper 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 codifies the string found with a two-number code composed by: the distance between the two strings and the length m of the string found. If the zipper does not find any match then it codifies the first character to be zipped, x_{n+1}, with its name. This eventuality happens for instance when codifying the first characters of the sequence, but this event becomes very infrequent as the zipping procedure goes on.

This zipper is asymptotically optimal: i.e. if it encodes a text of length L emitted by an ergodic source whose entropy per character is h, then the length of the zipped file divided by the length of the original file tends to h when the length of the text tends to ∞. The convergence to this limit is slow and the corrections has been shown to behave as $\mathrm{O}(\frac{\log\log L}{\log L})$ [31].

J. Stat. Mech. (2005) P04002

Usually, in commercial implementations of LZ77 (for instance *gzip*), substitutions are made only if the two identical sequences are not separated by more than a certain number n_w of characters, and the zipper is said to have an n_w-long sliding window. The typical value of n_w is 32 768. The main reason for this restriction is that the search in very large buffers could be inefficient from the computational time point of view.

Just to give an example: if one compresses an English text the length of the zipped file is typically of the order of a quarter of the length of the initial file. An English file is encoded with 1 byte (8 bits) per character. This means that after the compression the file is encoded with about 2 bits per character. Obviously this is not yet optimal. Shannon with an ingenious experiment showed that the entropy of the English text is between 0.6 and 1.3 bits per character [32] (for a recent study see [19]).

It is well known that compression algorithms represent a powerful tool for the estimation of the AC or more sophisticated measures of complexity [33]–[37] and several applications have been found in several fields [38] from dynamical systems theory (the connections between information theory and dynamical systems theory are very strong and go back all the way to Kolmogorov and Sinai [39, 40]; for a recent overview see [41]–[43]) to linguistics (an incomplete list would include [3], [6]–[9], [44]–[48]), genetics (see [5, 10], [49]–[52] and references therein) and music classification [53, 54].

2.1. Remoteness between two texts

It is interesting to recall the notion of relative entropy (or Kullback–Leibler divergence [55]–[57]) which is a measure of the statistical remoteness between two distributions and whose essence can be easily grasped with the following example.

Let us consider two stationary zero-memory sources $\mathcal{A}$ and $\mathcal{B}$ emitting sequences of 0 and 1: $\mathcal{A}$ emits a 0 with probability p and 1 with probability $1-p$ while $\mathcal{B}$ emits 0 with probability q and 1 with probability $1-q$. As already described, a compression algorithm like LZ77 applied to a sequence emitted by $\mathcal{A}$ will be asymptotically (i.e. in the limit of an available infinite sequence) able to encode the sequence almost optimally, i.e. coding on average every character with $-p\log_2 p-(1-p)\log_2(1-p)$ bits (the Shannon entropy of the source). This optimal coding will not be the optimal one for the sequence emitted by $\mathcal{B}$. In particular the entropy per character of the sequence emitted by $\mathcal{B}$ in the coding optimal for $\mathcal{A}$ (i.e. the cross-entropy per character) will be $-q\log_2 p-(1-q)\log_2(1-p)$ while the entropy per character of the sequence emitted by $\mathcal{B}$ in its optimal coding is $-q\log_2 q-(1-q)\log_2(1-q)$. The number of bits per character wasted in encoding the sequence emitted by $\mathcal{B}$ with the coding optimal for $\mathcal{A}$ is the relative entropy of $\mathcal{A}$ and $\mathcal{B}$,

$$d(\mathcal{A}||\mathcal{B}) = -q\log_2\frac{p}{q}-(1-q)\log_2\frac{1-p}{1-q}. \tag{5}$$

A linguistic example will help to clarify the situation: transmitting an Italian text with a Morse code optimized for English will result in the need for transmitting an extra number of bits with respect to another coding optimized for Italian: the difference is a measure of the relative entropy of, in this case, Italian and English (supposing the two texts are each archetypal representations of their language, which they are not).

We should remark that the relative entropy is not a distance (metric) in the mathematical sense: it is neither symmetric, nor does it satisfy the triangle inequality.

J. Stat. Mech. (2005) P04002

As we shall see below, in many applications, such as phylogenesis, it is crucial to define a true metric that measures the actual distance between sequences.

There exist several ways to measure the relative entropy (see for instance [35]–[37]). One possibility is of course to follow the recipe described in the previous example: using the optimal coding for a given source to encode the messages of another source.

Here we follow the approach recently proposed in [3] which is similar to the approach of Ziv and Merhav [36]. In particular in order to define the relative entropy of two sources $\mathcal{A}$ and $\mathcal{B}$ we consider a sequence A from the source $\mathcal{A}$ and a sequence B from the source $\mathcal{B}$. We now perform the following procedure. We create a new sequence $A+B$ by appending B after A and use the LZ77 algorithm or, as we shall see below, a modified version of it.

In [11] there was a detailed study of what happens when a compression algorithm tries to optimize its features at the interface between two different sequences A and B while zipping the sequence $A+B$ obtained by simply appending B after A. In particular the existence of a scaling function ruling the way the compression algorithm learns a sequence B after having compressed a sequence A has been shown. In particular it turns out that there exists a crossover length for the sequence B, given by

$$L_B^* \simeq L_A^\alpha \tag{6}$$

with $\alpha = h(B)/(h(B)+d(B||A))$. This is the length below which the compression algorithm does not learn the sequence B (measuring in this way the cross-entropy of A and B) and above which it learns B, i.e. optimizes the compression using the specific features of B.

This means that if B is short enough (shorter than the crossover length), one can measure the relative entropy by zipping the sequence $A+B$ (using *gzip* or an equivalent sequential compression program); the measure of the length of B in the coding optimized for A will be $\Delta_{AB} = L_{A+B} - L_A$, where L_X indicates the length in bits of the zipped file X. The cross-entropy per character of $\mathcal{A}$ and $\mathcal{B}$ will be estimated by

$$C(\mathcal{A}|\mathcal{B}) = \Delta_{AB}/|B|, \tag{7}$$

where $|B|$ is the length in bits of the uncompressed file B. The relative entropy $d(\mathcal{A}||\mathcal{B})$ per character of $\mathcal{A}$ and $\mathcal{B}$ will be estimated by

$$d(\mathcal{A}||\mathcal{B}) = (\Delta_{AB} - \Delta_{B'B})/|B|, \tag{8}$$

where B' is a second sequence extracted from the source $\mathcal{B}$ with $|B'|$ characters and $\Delta_{B'B}/|B| = (L_{B+B'} - L_B)/|B|$ is an estimate of the entropy of the source $\mathcal{B}$.

If, on the other hand, B is longer than the crossover length we must change our strategy and implement an algorithm which does not zip the B part but simply 'reads' it with the (almost) optimal coding of part A. In this case we start reading sequentially file B and search in the look-ahead buffer of B for the longest subsequence that already occurred *only* in the A part. This means that we do not allow for searching matches inside B itself. As in the usual LZ77, every matching found is replaced with a pointer indicating where, in A, the matching subsequence appears and its length. This method allows us to measure (or at least to estimate) the cross-entropy of B and A, i.e. $C(\mathcal{A}|\mathcal{B})$.

Before proceeding let us briefly discuss what difficulties one could experiment on in the practical implementation of the methods described in this section. First of all in practical applications the sequences to be analysed can be very long and their direct comparison

J. Stat. Mech. (2005) P04002

can then be problematic due to finiteness of the window over which matching can be found. Moreover in some applications one is interested in estimating the self-entropy of a source, i.e. $C(\mathcal{A}|\mathcal{A})$, in a more coherent framework. The estimation of this quantity is necessary for calculating the relative entropy of two sources. In fact, as we shall see in the next section, even though in practical applications the simple cross-entropy is often used, there are cases in which the relative entropy is more suitable. The most typical case is when we need to build a symmetrical distance between two sequences. One could think of estimating self-entropy comparing, with the modified LZ77, two portions of a given sequence. This method is not very reliable since many biases could afflict the results obtained in this way. For example if we split a book into two parts and try to measure the cross-entropy of these two parts, the result we would obtain could be heavily affected by the names of the characters present in the two parts. More importantly, defining the position of the cut would be completely arbitrary, and this arbitrariness would matter a lot especially for very short sequences. We shall address this problem in section 3.

2.2. On the definition of a distance

In this section we address the problem of defining a distance between two generic sequences A and B. A distance D is an application that must satisfy three requirements:

(1) positivity: $D_{AB} \geq 0$ ($D_{AB} = 0$ iff $A = B$);
(2) symmetry: $D_{AB} = D_{BA}$;
(3) triangular inequality: $D_{AB} \leq D_{AC} + D_{CB} \forall C$.

As is evident, the relative entropy $d(\mathcal{A}||\mathcal{B})$ does not satisfy the last two properties while it is never negative. Nevertheless one can define a symmetric quantity as follows:

$$P_{AB} = P_{BA} = \frac{C(\mathcal{A}|\mathcal{B}) - C(\mathcal{B}|\mathcal{B})}{C(\mathcal{B}|\mathcal{B})} + \frac{C(\mathcal{B}|\mathcal{A}) - C(\mathcal{A}|\mathcal{A})}{C(\mathcal{A}|\mathcal{A})}. \tag{9}$$

We now have a symmetric quantity, but P_{AB} does not satisfy, in general, the triangular inequality. In order to obtain a real mathematical distance we give a prescription according to which this last property is met. For every pair A and B of sequences, the prescription is

$$\begin{aligned} &\text{if } P_{AB} > \min_C [P_{AC} + P_{CB}] \text{ then} \\ &P_{AB} = \min_C [P_{AC} + P_{CB}]. \end{aligned} \tag{10}$$

By iterating this procedure until for any A, B, C, $P_{AB} \leq P_{AC} + P_{CB}$, we obtain a true distance D_{AB}. In particular the distance obtained in this way is simply the minimum over all the paths connecting A and B of the total cost of the path (according to P_{AB}): i.e.,

$$D_{AB} = \min_{\{N \geq 2\}} \min_{\{X_1, \ldots, X_N : X_1 = A, X_N = B\}} \sum_{k=0}^{N-1} P_{X_k X_{k+1}}. \tag{11}$$

Also it is easy to see that D_{AB} is the maximal distance not larger than $P_{A,B}$ for any A, B, where we have considered a partial ordering on the set of distances: $P \geq P'$ if $P_{AB} \geq P'_{AB}$, for all pairs A, B.

Obviously this is not an *a priori* distance. The distance between A and B depends, in principle, on the set of files we are considering.

In all our tests with linguistic texts the triangle condition was always satisfied without the need to have recourse to the above-mentioned prescription. However there are cases in other contexts, for instance, genetic sequences, in which it could be necessary to force the triangularization procedure described above.

An alternative definition of distance can be given considering

$$R_{AB} = \sqrt{P_{AB}}, \tag{12}$$

where the square root must be taken before forcing the triangularization. The idea of using R_{AB} is suggested by the fact that as A and B are very close sources, P_{AB} is of the order of the square of their 'difference'. Let us see this in a concrete example where the distance between the two sources is very small. Suppose we have two sources $\mathcal{A}$ and $\mathcal{B}$ which can emit sequences of 0 and 1. Let $\mathcal{A}$ emit a 0 with a probability p and 1 with the complementary probability $1-p$. Now let the source $\mathcal{B}$ emit a 0 with a probability $p+\epsilon$ and a 1 with a probability $1-(p+\epsilon)$, where ϵ is an infinitesimal quantity. In this situation it can be easily shown that the relative entropy of $\mathcal{A}$ and $\mathcal{B}$ is proportional to ϵ^2 and, of course, P_{AB} is then proportional to the same quantity. Taking the square root of P_{AB} is then simply requiring that, if two sources have a distribution of probability that differs for a small ϵ, their distance must be of the order of ϵ instead of being reduced to the ϵ^2 order.

It is important to recall that an earlier and rigorous definition of an unnormalized distance between two generic strings of characters has been proposed in [58] in terms of the Kolmogorov complexity and of the conditional Kolmogorov complexity [30] (see below for the definition).

A normalized version of this distance has been proposed in [52, 59]. In particular Li *et al* define

$$d_K(x,y) = \frac{\max(K(x|y), K(y|x))}{\max(K(x), K(y))} \tag{13}$$

where the subscript K refers to its definition in terms of the Kolmogorov complexity. $K(x|y)$ is the conditional Kolmogorov complexity defined as the length of the shortest program for computing x if y is furnished as an auxiliary input to the computation, and $K(x)$ and $K(y)$ are the Kolmogorov complexities of strings x and y, respectively. The distance $d_K(x,y)$ is symmetrical and it is shown to satisfy the identity axiom up to a precision $d_K(x,x) = \mathrm{O}(1/K(x))$ and the triangular inequality $d_K(x,y) <= d_K(y,z) + d_K(z,y)$ up to an additive term $\mathrm{O}(1/\max(K(x), K(y), K(z)))$.

The problem with this distance is the fact that it is defined in terms of the conditional Kolmogorov complexity which is an uncomputable quantity and its computation is performed in an approximate way.

In particular what is important is that the specific procedure (algorithm) used to approximate this quantity, which is indeed a well defined mathematical operation, defines a true distance. In the specific case of the distance $d_K(x,y)$ defined in [52] the authors approximate this distance by the so-called normalized compression distance

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))} \tag{14}$$

where $C(xy)$ is the compressed size of the concatenation of x and y, and $C(x)$ and $C(y)$ denote the compressed sizes of x and y, respectively. Then these quantities are approximated in a suitable way by using real world compressors.

It is important to remark that there exists a discrepancy between the definition (13) and its actual approximate computation (14).

We discuss here in some detail the case of the LZ77 compressor. Using the results presented in section 2.1, one obtains that, if the length of y is small enough (see expression (6)), $\mathrm{NCD}(x, y)$ is actually estimating the cross-entropy of x and y. The cross-entropy is not a distance since it does not satisfy the identity axiom, it is not symmetrical and it does not satisfy the triangular inequality. In the general case of y being not small, again following the discussion of section 2.1 (presented in more detail in [11]), one can show that $\mathrm{NCD}(x, y)$ is given roughly (for L_x large enough) by

$$1 + \frac{L_x^{\alpha}}{L_y} \frac{d(x||y)}{C(y)}, \tag{15}$$

where L_x and L_y are the lengths of the x and y files (with $L_y \gg L_x^{\alpha}$) and $d(x||y)$ is the relative entropy rate of x and y. Again this estimate does not define a metric. Moreover, since $\alpha \leq 1$ one can see that $\mathrm{NCD}(x, y) \to 1$, independently of the choice of x and y when L_x and L_y tend to infinity.

The discrepancy between the definition of a mathematical distance based on the conditional Kolmogorov complexity and its actual approximate computation in [52] has also been pointed out in [60].

Finally it is important to note that recently Otu and Sayood [61] have proposed an alternative definition of distance between two strings of characters, which is rigorous and computable. Their approach is based on the LZ complexity [62] of a sequence S which can be defined in terms of the number of steps required by a suitable production process to generate S. In their very interesting paper they also give a review on this and correlated problems. We do not enter here into details and we refer the reader to [61].

3. Dictionaries and artificial texts

As we have seen, LZ77 replaces sequences of characters with a pointer to their previous appearance in the text. We now need some definitions before proceeding. We describe as a *dictionary* of a sequence the whole set of subsequences replaced with a pointer by LZ77, and we refer to these sequences as the dictionary's *words*. As is evident from these definitions, a particular word can be present many times in the dictionary. Finally, we describe as a *root* of a dictionary the sequence it has been extracted from. It is important to stress how this dictionary has in principle nothing to do with the ordinary dictionary of a given language. On the other hand there could be important similarities of the LZ77 dictionary of a written text and the dictionary of the language in which the text is written. As examples, we report in tables 1 and 2 the most frequent and the longest *words* found by LZ77 while zipping Melville's Moby Dick text. Figure 2 reports an example of the frequency–length distribution of the LZ77 words as a function of their length (for a very similar figure and similar but less complete dictionary analysis see [10]).

Beyond their utility for zipping purposes, the dictionaries have intrinsic interest since one can consider them as a source for the principal and more important syntactic structures present in the sequence/text from which the dictionary originates.

J. Stat. Mech. (2005) P04002

Table 1. Most frequent LZ77 words found in Moby Dick's text: here we present the most represented word in the dictionary of Moby Dick. The dictionary was extracted using a 32 768 sliding window in LZ77. The ⌣ represents the space character.

Frequency	Length	Word
110	6	.⌣The⌣
107	7	in⌣the⌣
98	4	you⌣
94	6	.⌣But⌣
92	9	from⌣the⌣
92	5	⌣very⌣
91	4	one⌣

Table 2. Longest words in Moby Dick: here we present the longest words in the dictionary of Moby Dick. Each of these words appears only one time in the dictionary. The dictionary was extracted using a 32 768 sliding window in LZ77.

Frequency	Length	Word
1	80	,–⌣Such⌣a⌣funny, ⌣sporty, ⌣gamy, ⌣jesty, ⌣joky, ⌣hoky–poky⌣lad, ⌣is⌣the⌣Ocean, ⌣oh!⌣Th
1	78	,–⌣Such⌣a⌣funny, ⌣sporty, ⌣gamy, ⌣jesty, ⌣joky, ⌣hoky–poky⌣lad, ⌣is⌣the⌣Ocean, ⌣oh!⌣
1	63	'⌣' I⌣look, ⌣you⌣look, ⌣he⌣looks; ⌣we look, ⌣ye⌣look, ⌣they look.'⌣'W
1	63	'!⌣' I⌣look, ⌣you⌣look, ⌣he⌣looks; ⌣we look, ⌣ye⌣look, ⌣they look. '⌣'
1	54	repeated⌣in⌣this⌣book, ⌣that⌣the the⌣skeleton⌣of⌣the whale
1	46	.⌣THIS⌣TABLET⌣Is⌣erected⌣to⌣his ⌣Memory⌣BY⌣HIS⌣
1	43	s⌣a⌣mild, ⌣mild⌣wind, ⌣and⌣a⌣mild⌣looking⌣sky

A straightforward application is the possibility of constructing *artificial texts*. By this name we mean sequences of characters built by concatenating words randomly extracted from a specific dictionary.

Each word has a probability of being extracted proportional to the number of its occurrences in the dictionary. Since typically LZ77 words already contain spaces, we do not include further spaces separating them. It should be stressed as the structure of a dictionary is affected by the size of LZ77 sliding window. In our case we have typically adopted windows of 32 768 characters, and, in a few cases, of 65 536 characters.

Below we present an excerpt of 400 characters taken from an artificial text (AT) having Melville's Moby Dick text as the root.

```
those boats round with at coneedallioundantic turneeling he had
Queequeg, man.''Tisheed the o corevolving se were by their fAhab tcandle
aed.  Cthat the ive ing, head upon that can onge Sirare ce more le in and
for contrding to the nt him hat seemed ore, es; vacaknowt.''  ''  it
```

```
seemside delirirous from the gan.  All ththe boats bedagain, brightflesh,
yourselfhe blacksmith's leg t.  Mre?loft restoon
```

As is evident the meaning is completely lost and the only feature of this text is to represent in a significant statistical way the typical structures found in the original root text (i.e. the typical subsequences of characters).

The case of sequences representing texts is interesting, and it is worth spending a little time on it, since a clear definition of 'word' already exists in every language. In this case one could also define *natural* artificial texts (NAT). A NAT is obtained by concatenating true words as extracted from a specific text written in a certain language. Also in this case each word would be chosen according to a probability proportional to the frequency of its occurrence in the text. Just for comparison with the previous AT we report an example of a natural artificial text built using real words from the English dictionary taken randomly with a probability proportional to their frequency of occurrence in Moby Dick's text.

```
of Though sold, moody Bedford opened white last on night; FRENCH
unnecessary the charitable utterly form submerged blood firm-seated
barricade, and one likely keenly end, sort was the to all what ship nine
astern; Mr.  and Rather by those of downward dumb minute and are essential
were baby the balancing right there upon flag were months, equatorial
whale's Greenland great spouted know Delight, had
```

We now describe how artificial texts can be effectively used for recognition and classification purposes. First of all AT present several positive features. They allow one to define typical *words* for generic sequences (not only for texts). Moreover for each original text (or original sequence), one can construct an *ensemble* of AT. This opens the way to the possibility of performing statistical analysis by comparing the features of many AT all representative of the same original root text. In this way it is possible to overcome all the difficulties, discussed in the previous section, related to the length of the strings analysed. In fact it seems very plausible that, once a certain 'reasonable' AT size has been established, any string can be well represented by a number of AT proportional to its length. On the other hand one can construct AT by merging dictionaries coming from different original texts: merging dictionaries extracted from different texts all about the same subject or all written by the same author. In this way the AT would play the role of an archetypal text of that specific subject or that specific author [63].

The possibility of constructing many different AT all representative of the same original sequence (or of a given source) allows for an alternative way to estimate the self-entropy of a source (and consequently the relative entropy of two sources as mentioned above). The cross-entropy of two AT corresponding to the same source will give in fact directly an estimate of the self-entropy of the source. This is an important point since in this way it is possible to estimate the relative entropy and the distances between two texts of the form proposed in equation (9) in a coherent framework. Finally, as is shown in figure 3, comparing many AT coming from the same two roots (or a single root), we can estimate a statistical error on the value of the cross-entropy of the two roots.

With the help of AT we can then build a comparison scheme (artificial text comparison or ATC; see figure 3) between sequences whose validity will be checked in the following sections. This scheme is very general since it can be applied to any kind of sequence independently of the coding behind it. Moreover the generality of the scheme comes from the fact that, by means of a redefinition of the concept of 'word', we are able to

J. Stat. Mech. (2005) P04002

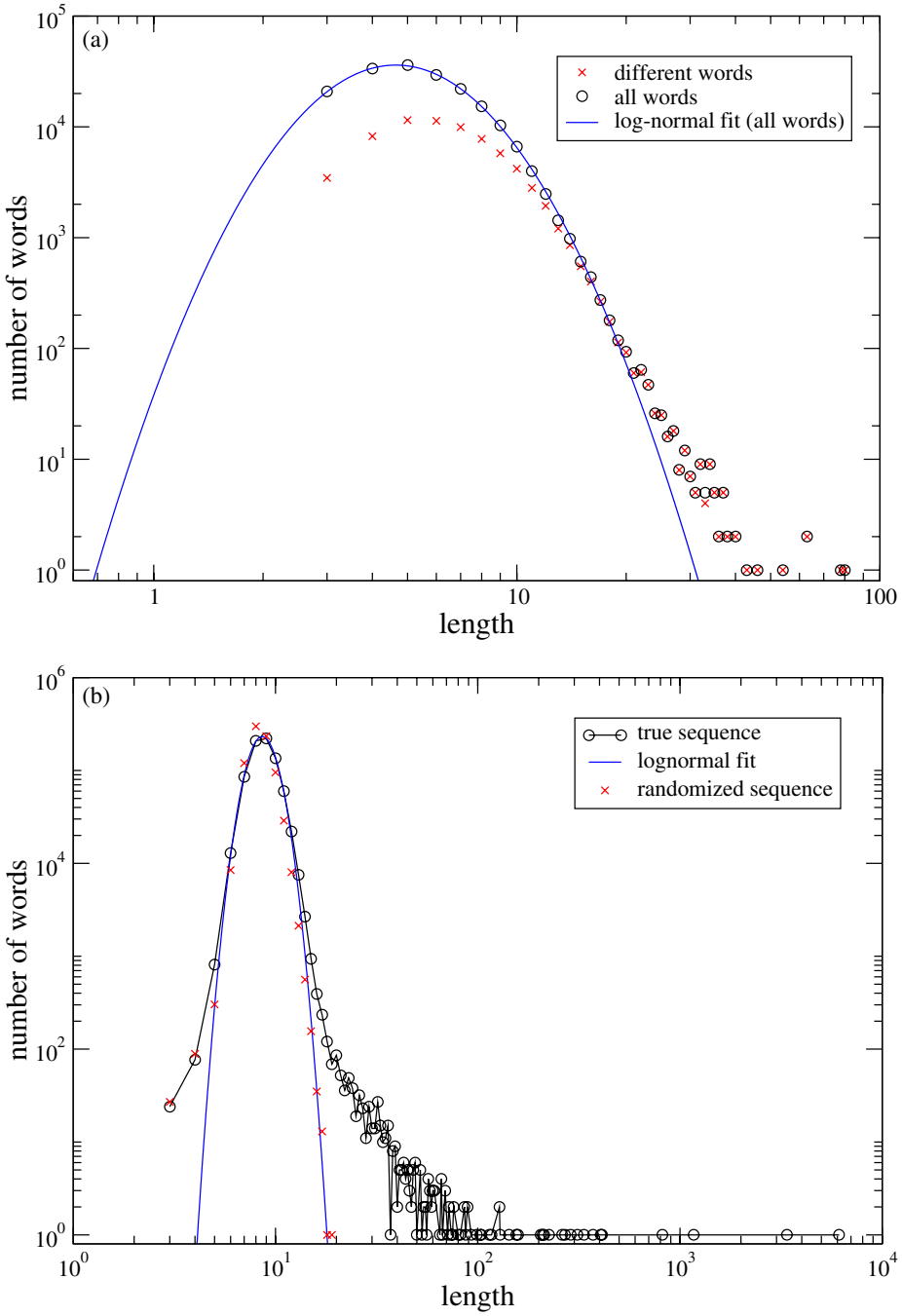

Figure 2. LZ77 word distribution: this figure illustrates the distribution of the LZ77 words found in different strings of characters. Above: results for the dictionary of *Moby Dick* are shown. In the upper curve several findings of the same word are considered separately; in the lower curve each different word is counted only once. It can be shown that the peaks are well fitted by a log-normal distribution, while there are large deviations from it for large lengths. Below: words extracted from *Mesorhizobium loti* bacterium's original and reshuffled DNA sequences are analysed. The log-normal curve fits well the whole distribution of words extracted from the reshuffled string, but is unable to describe the presence of the long words of the true one.

J. Stat. Mech. (2005) P04002

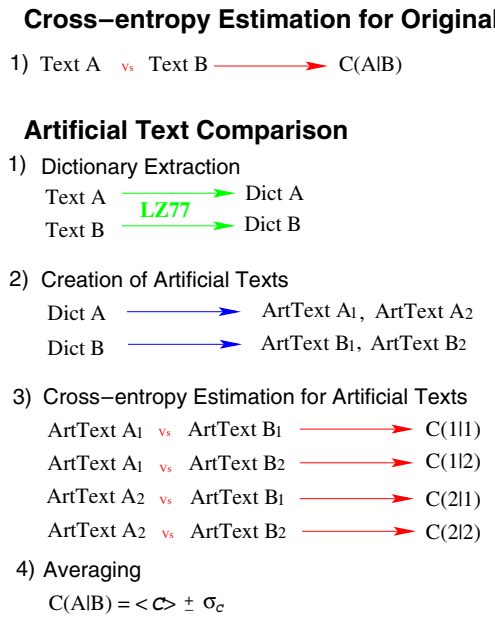

Figure 3. Artificial text comparison (ATC) method: this is the scheme of the artificial text comparison method. Instead of comparing two original strings, several AT (two in figure) are created starting from the dictionaries extracted from the original strings, and the comparison is between pairs of AT. For each pair of AT coming from different roots a cross-entropy value $C(i|j)$ is measured and the cross-entropy of the root strings is obtained as the average $\langle C\rangle$ of all the $C(i|j)$. This method has the advantage of allowing for an estimation of an error, σ, on the value obtained for the cross-entropy $\langle C\rangle$, as the standard deviation of the $C(i|j)$. From the point of view of the ATC computational demand, point (1) simply consists in the procedure of zipping the original files, that usually requires a few seconds, points (2) and (4) are of course negligible, while point (3) is crucial. Obviously, in fact, the machine time required for the cross-entropy estimation grows as the square power of the number of AT created (for fixed length of the AT).

extract subsequences from a generic sequence using a deterministic algorithm (for instance LZ77) which eliminates every arbitrariness (at least once the algorithm for the dictionary extraction has been chosen). In the following sections we shall discuss in detail how one can use AT for recognition and classification purposes.

4. Recognition of linguistic features

Our first experiments are concerned with recognition of linguistic features. Here we consider those situations in which we have a corpus of *known* texts and one unknown text X. We are interested here in identifying the known text A closest (according to some rule) to the X one. We then say that X, being similar to A, belongs to the same group as A. This group can, for instance, be formed by all the works of an author, and in that case we say that our method attributed X to that author. We now present results obtained in experiments on language recognition and authorship attribution. After having explained

our experiments we will be able to make some more comments on the criterion we adopted to set recognition and/or attribution.

4.1. Language recognition

Suppose we are interested in the automatic recognition of the language in which a given text X is written. This case can be seen as a first benchmark for our recognition technique. The procedure we use considers a collection (a corpus), as large as possible, of texts in different (known) languages: English, French, Italian, Tagalog,.... We take an X text to play the role of the unknown text whose language has to be recognized, and the remaining A_i texts of our collection to form our background. We then measure the cross-entropy of our X text and every A_i with the procedure discussed in section 2. The text among the A_i group with the smallest cross-entropy with the X one selects the language closest to that of the X file, or exactly its language, if the collection of languages contains this language. In our experiment we have considered in particular a corpus of texts in 10 official languages of the European Union (UE) [64]: Danish, Dutch, English, Finnish, French, German, Italian, Portuguese, Spanish and Swedish. Using 10 texts for each language we had a collection of 100 texts. We have obtained that for any single text the method has recognized the language. This means that the text A_i for which the cross-entropy with the unknown X text was the smallest was a text written in the same language. We found also that if we ranked for each X text all the texts A_i as a function of the cross-entropy, all the texts written in the same language of the unknown text were in the first positions. This means that the recall, defined in the framework of information retrieval as the ratio of the number of relevant documents retrieved (independently of the position in the ranking) and the total number of existing relevant documents, is maximal, i.e. equal to one. The recognition of language works quite well for lengths of the X file as small as a few tens of characters.

4.2. Authorship attribution

Suppose now that we are interested in the automatic recognition of the author of a given text X. We shall consider, as before, a collection, as large as possible, of texts of several (known) authors all written in the same language as the unknown text and we shall look for the text A_i for which the cross-entropy with the X text is minimum. In order to collect a certain level of statistics we have performed the experiment using a corpus of 87 different texts [65] by 11 Italian authors, using for each run one of the texts in the corpus as the unknown X text. In a first step we proceeded exactly as for language recognition, using the actual texts. The results, shown in table 3, feature a rate of success of roughly 93%. This rate is the ratio of the number of texts whose author has been recognized (another text by the same author was ranked as first) and the total number of texts considered. There are of course fluctuations in the success rate for authors and this has to be expected since the writing style is something difficult to grasp and define; moreover it can vary a lot in the production of a single author.

We then proceeded to analysing the same corpus with the ATC method we have discussed in the previous section. We extracted the dictionary from each text, and built up our 87 artificial texts (each one 30 000 characters long). In each run of our experiment we chose one artificial text to play the role of the text whose author was unknown and

Table 3. Author recognition: this table illustrates the results for the experiments on author recognition. For each author we report the number of different texts considered and a measure of success for each of the three methods adopted. Labelled as successes are the numbers of times another text by the same author was ranked in the first position using the minimum cross-entropy criterion.

Author	Number of texts	Successes: actual texts	Successes: ATC	Successes: NATC
Alighieri	5	5	5	5
D'Annunzio	4	4	4	4
Deledda	15	15	15	15
Fogazzaro	5	4	5	5
Guicciardini	6	5	6	6
Machiavelli	12	12	11	10
Manzoni	4	3	4	4
Pirandello	11	11	11	11
Salgari	11	10	11	11
Svevo	5	5	5	5
Verga	9	7	9	9
Totals	87	81	86	85

the other 86 to be our background. The result is significant. We found that 86 times out of 87 trials the author was indeed recognized, i.e. the cross entropy of our unknown text and at least another text by the right author was the smallest. This means that the rate of success using artificial texts was of 98.8%. The unrecognized text was *L'Asino* by Machiavelli, which was attributed to Dante (*La Divina Commedia*), and, in fact, these are both poetic texts; so it does not appear so strange thinking that *L'Asino* is found to be in some way closer to the *Commedia* than to *Il Principe*. A slightly different way of proceeding is the following. Instead of extracting an artificial text from each actual text, we made a single artificial text, which we call the *author archetype*, for each author. To do this we simply joined all the dictionaries for the author and then proceeded as before. In this case we used actual works as unknown texts and author archetypes as background. We obtained that 86 out of 87 unknown real texts matched the right artificial author text, the one missing being again *L'Asino*.

In order to investigate this mismatching further we exploited one of the biggest advantages the ATC method can give if compared to real text comparison. While in real text comparison only one trial can be made, ATC allows for creating an ensemble of different artificial texts, and so more than one trial is possible. In our specific case, however, 10 different ATC trials performed both with artificial texts and with author archetypes gave the same result, attributing *L'Asino* to Dante. This can probably confirm our supposition that the pattern of poetic register is very strong in this case. To be sure that our 98.8% rate of success was not due to a particular fortuitous accident in our set of artificial texts, we repeated our experiment with a corpus formed by 5 artificial texts from each actual text. This means that our collection was formed by 435 texts. We then proceeded in the usual way. Having our cross-entropies of the $5X_n$ ($n = 1, \ldots, 5$) artificial texts coming from the same root X, and the remaining 430 ATs, we first joined all the rankings relative to these X_n. Thus we had 430×5 cross-entropies of the AT extracted by

the same root X and the other AT of our ensemble. We then averaged, for each root A_i, all the 25 cross-entropies of an AT created from X text and an AT extracted from that A_i. In this way we obtained 86 cross-entropy values, and we set authorship attribution using the usual minimum criterion. We found again that 86 texts out of 87 were well attributed, *L'Asino* being again misattributed.

This result shows that ATC is a robust method since it does not seem to be strongly influenced by the particular set of artificial texts. In particular, as we have discussed before, ATC allows for a quantification of the error in the cross-entropy estimation. Defined as σ_m, the standard deviation estimated for the mth cross-entropy, in a ranking in which the smallest cross-entropy value is the first one, we empirically observed these relations:

$$\frac{\sigma_1}{C_1} \simeq \frac{\sigma_2}{C_2} \simeq \frac{\sigma_3}{C_3} \simeq 0.5\% \tag{16}$$

$$(C_2 - C_1) \simeq \sigma_1 \simeq \sigma_2. \tag{17}$$

The difference $C_2 - C_1$ gives an indication of the level of confidence of the results. When this difference is of the order of the standard deviation of C_1 and C_2, this is an indication that the result for the attribution has a high level of confidence (at least inside the corpus of reference files/texts considered).

Finally, in order to explore the possibility of using natural words, we performed experiments with natural artificial texts. We call this method Natural ATC or NATC. We built up five artificial texts for each actual one using Italian words instead of words extracted by LZ77. Having these natural artificial texts we proceeded exactly as before. We obtained that 85 out of 87 texts were recognized. Besides *L'Asino*, the other mismatch was the *Istorie Fiorentine* by Machiavelli that was set closest to *Storie Fiorentine dal 1378 al 1509* by Guicciardini. It seems clear that the closeness of the subjects treated in the two texts played a fundamental role in the attribution.

It is interesting trying out some conjectures on why artificial texts made up by the LZ77 extracted dictionary worked better in our experiment. Probably the main reason is that LZ77 very often puts some correlation between characters and actual words by grouping them into a single *word*, while clearly this correlation does not exist using natural words. In a text written to be read, words and/or characters are correlated in a precise way, especially in some cases (one of the most strict, but probably less significant, is '.' followed by a capital letter). These observations could maybe suggest that LZ77 is able to capture correlations that are in some sense a signature of an author, this signature being stronger (up to a certain point, of course) than that of the subject of a particular text. On the other hand this ability of keeping a memory of correlations, combined with the specificity of the poetic register, could also explain the apparent strength of the poetic pattern that seems to emerge from our experiments.

We have also performed some additional experiments on a corpus of English texts. Results are shown in table 4. In this corpus there were a few poetic texts which, as we could expect, were problematic for ATC. It is worth noting, in fact, that the number of ATC failures is 7, and in this case it is higher than the value for actual text comparisons, which is 4. However, if we look carefully we note that four of these seven mismatches come from the five Marlowe works present in our corpus. Among Marlowe's works only one is

J. Stat. Mech. (2005) P04002

Table 4. Author recognition: this table illustrates the results for the experiments on author recognition. In this case ATC results were afflicted by the presence in the corpus of a few poetic texts that, as we have discussed, tend to recognize each other.

Author	Number of texts	Successes: actual texts	Successes: ATC	Successes: NATC
Bacon	6	6	6	6
Brown	3	2	2	2
Chaucer	6	6	6	6
Marlowe	5	4	1	2
Milton	8	8	7	7
Shakespeare	37	37	37	37
Spencer	7	5	6	5
Totals	72	68	65	65

misattributed by actual text comparison, too. This peculiarity of Marlowe roused our interest and we analysed carefully Marlowe's results. We found that one of the four bad attributions was a poetic text, *Hero*, and was attributed to Spencer, while the remaining three unrecognized texts were all attributed to Shakespeare. Similar results were obtained using the NATC method which also does not allow for a clear distinction between Marlowe and Shakespeare. Just as a matter of curiosity, and without entering into the debate, we report here that, among the many theses on the real identity of Shakespeare, there is one that claims that Shakespeare was just a pseudonym used by Marlowe to sign some of his works. The Marlowe Society embraces this cause and has presented many works which could prove this theory, or at least make it plausible (starting of course by refuting the official date of death of Marlowe, 1593).

Before concluding this section several remarks are in order concerning our minimum cross-entropy method used to perform authorship attribution. Our criterion has been that of saying that the X should be attributed to a given author if another work of this author is the closest (in the cross-entropy ranking) to X. It can happen, and sometimes this is the case, that the text second closest to X belongs to another author, different from the first. In other words, in the ranking of relative entropies of the X text and all the other texts of our corpus, works belonging to a given author are far from clustering in the same zone of the ranking. This fact can be easily explained with the large variety of features that can be present in the production of an author. Dante, for instance, wrote both poetry and prose, the latter both in Italian and in Latin. In order to take into account this non-homogeneity we decided to set authorship by watching only at the closest text to the unknown one. In fact, from what we have said, averaging or taking into account all the texts of every author could introduce biases given by the heterogeneity in each author's production. Our choice is then perfectly coherent with the purpose of authorship attribution which is not to determine an *average* author of the unknown text, but who wrote that particular text. The limitation of this method is the assumption that if an author wrote a text, then he or she is likely to have written a similar text, at least as regards structural or syntactic aspects. From our experiments we can say, *a posteriori*, that this assumption does not seem to be unrealistic.

J. Stat. Mech. (2005) P04002

A further remark concerns the fact that our results for authorship attribution could only provide some hints about the real paternity of a text. One cannot, in fact, ever be sure that the reference corpus contains at least one text by the unknown author. If this is not the case we can only say that some works of a given author resemble the unknown text. On the other hand the method could be highly effective when one has to decide among a limited and predefined set of candidate authors: see for instance the *Wright–Wright* problem [66] and the *Grunberg–Van der Jagt* problem in The Netherlands [67].

From a general point of view, finally, it is important to remark that the ATC method is of much greater interest than the NATC one. In fact, even though in linguistic related problems the two methods give comparable results, ATC can be used with every set of generic sequences, while the NATC requires a precise definition of words in the original strings.

5. Self-consistent classification

In this section we are interested in the classification of large corpora in situations where no *a priori* knowledge of the corpora's structure is given. Our method, mutuated by the phylogenetic analysis of biological sequences [68]–[70], considers the construction of a distance matrix, i.e. a matrix whose elements are the distances between pairs of texts. Starting from the distance matrix one can build a tree representation: phylogenetic trees [70], spanning trees etc. With these trees a classification is achieved by observing clusters that are supposed to be formed by similar elements. The definition of a distance between two sequences of characters has been discussed in section 2.2.

5.1. Author trees

In our applications we used the Fitch–Margoliash method [71] of the package PhylIP (Phylogeny Inference Package) [72] which basically constructs a tree by minimizing the net disagreement between the matrix pairwise distances and the distances measured on the tree. Similar results have been obtained with the 'Neighbor' algorithm [73]. The first test for our method consisted in analysing with the Fitch–Margoliash procedure the distance matrix obtained for the corpus of Italian texts used before for authorship attribution. Results are presented in figure 4. As can be seen, works by the same author tend to cluster quite well in the tree presented.

5.2. Language trees

The next step was applying our method in a less obvious context: that of a relationship between languages. Suppose we have a collection of texts written in different languages. More precisely, imagine we have a corpus containing several versions of the same text in different languages, and are interested in a classification of this corpus. In order to have the largest possible corpus of texts in different languages, we have used 'The Universal Declaration of Human Rights' [74] which sets the Guinness World Record for the most translated document.

We proceeded here for our analysis exactly as for author trees. We analysed with the Fitch–Margoliash method [71] the distance matrix obtained using the artificial text comparison method with five artificial texts for each real text. After averaging over

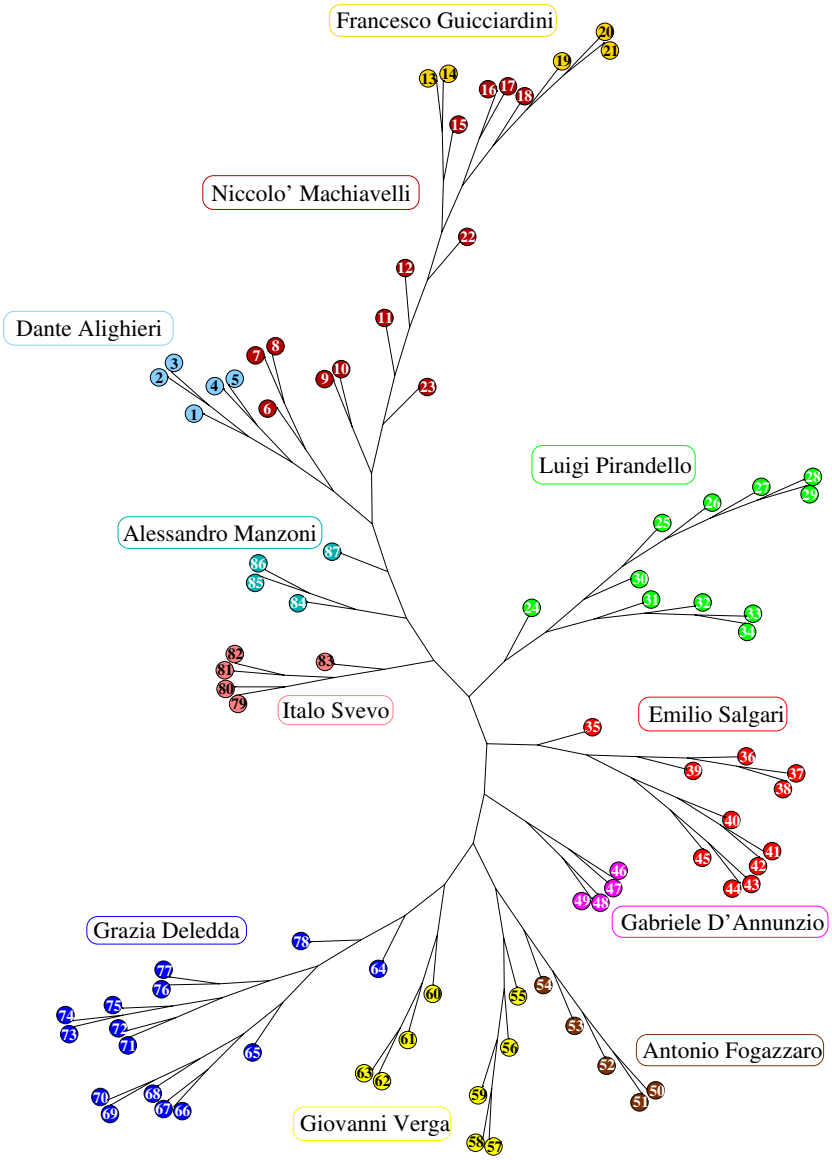

Figure 4. Italian authors' tree: the tree obtained with the Fitch–Margoliash algorithm using the P pseudo-distance built from the ATC method for the corpus of Italian texts considered in section 4.2. For the sake of clarity in the representation we have chosen a constant length for the distances between nodes and between nodes and leaves.

the artificial texts sharing the same root, we built up the distance matrix as discussed in section 2.2. In figure 5 we show the tree obtained with the Fitch–Margoliash algorithm for over 50 languages widespread on the Euro-Asiatic continent. We can notice that essentially all the main linguistic groups (Ethnologue source [75]) are recognized: Romance, Celtic, Germanic, Ugro-Finnic, Slavic, Baltic, Altaic. On the other hand one has isolated languages such as the Maltese, typically considered a Semitic language because of its Arabic base, and the Basque, a non-Indo-European language whose origins and relationships with other languages are uncertain. The results are also in good agreement

J. Stat. Mech. (2005) P04002

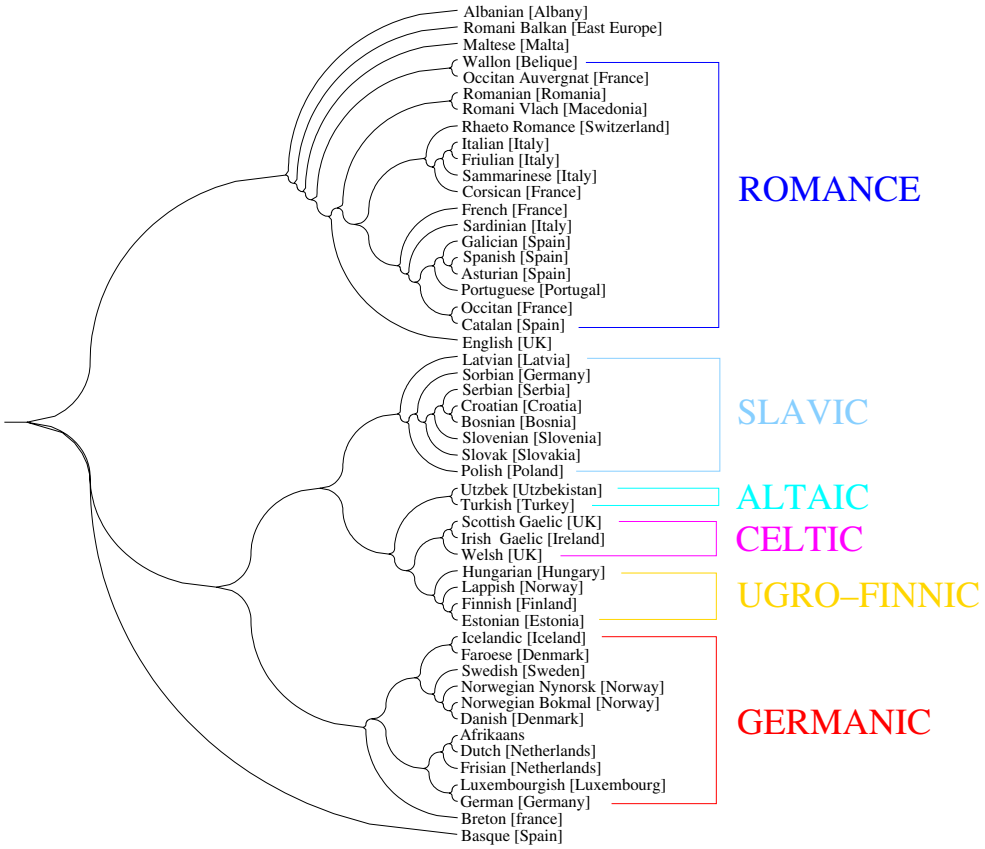

Figure 5. Indo-European family language tree: this figure illustrates the phylogenetic-like tree constructed on the basis of more than 50 different versions of the 'The Universal Declaration of Human Rights'. The tree is obtained using the Fitch–Margoliash method applied to the symmetrical distance matrix based on the R distance defined in section 2.2 built from the ATC method. This tree features essentially all the main linguistic groups of the Euro-Asiatic continent (Romance, Celtic, Germanic, Ugro-Finnic, Slavic, Baltic, Altaic), as well as a few isolated languages as the Maltese, typically considered an Afro-Asiatic language, and the Basque, classified as a non-Indo-European language and whose origins and relationships with other languages are uncertain. The tree is unrooted, i.e. it does not require any hypothesis about common ancestors for the languages and it cannot be used to infer information about common ancestors of the languages. For more details, see the text. The lengths of the paths between pairs of documents measured along the tree branches are not proportional to the actual distances between the documents.

with those obtained by true sequence comparison reported in [3], with a remarkable difference concerning the Ugro-Finnic group here fully recognized, while with true texts Hungarian was put a little apart.

After the publication of our tree in [3] a similar tree, using the same data set, was proposed in [52] using NCD(x, y) (see section 2.2) estimated with gzip.

It is important to stress that these trees are not intended to reproduce the current trends in the reconstruction of genetic relations among languages. They are clearly biased

J. Stat. Mech. (2005) P04002

by using entire modern texts for their construction. In the reconstruction of genetic relationships among languages one is typically faced with the problem of distinguishing *vertical* (i.e. the passage of information from parent languages to child languages) from *horizontal* transmission (i.e. which includes all the other pathways in which two languages interact). This is the main problem of lexicostatistics and glottochronology [76] and the most widely used method is that of the so-called Swadesh 100-word lists [77]. The main idea is that of comparing languages by comparing lists of so-called basic words. These lists only include the so-called cognate words, ignoring as much as possible horizontal borrowings of words between languages. It is clear now how an obvious source of bias in our results is represented by the fact of not having performed any selection of words to be compared. It turns out then that in our trees English is closer to Romance languages simply because almost 50% of English vocabulary has been borrowed from French. These borrowings should be expunged if one is interested in reconstructing the actual genetic relationships between languages. Work is currently in progress in an effort to merge Swadesh list techniques with our methods [78].

6. Discussion and conclusions

We have presented here a class of methods, based on the LZ77 compression algorithm, for information extraction and automatic categorization of generic sequences of characters. The essential ingredient of these methods is the definition and the measuring of remoteness and of the distance between pairs of sequences of characters. In this context we have introduced in particular the notion of a *dictionary* of a sequence and of an *artificial text* (or *artificial sequence*) and we have implemented these new tools in an information extraction scheme (ATC) that allows us to overcome several difficulties arising in the comparison of sequences.

With these tools in our hands, we have focused our attention on several applications to textual corpora in several languages, since in this context it is particularly easy to judge experimental results. We first showed that dictionaries are intrinsically interesting and that they contain relevant signatures of the texts they are extracted from. Then in a first series of experiments we have shown how we can determine, and then extract, some *semantic* attributes of an unknown text (its language, author or subject). We have also shown that comparing artificial texts, instead of actual sequences, gives better results in most of these situations. In the linguistic context, moreover, we have been able to define natural artificial texts (NAT) exploiting the presence of natural language words in the texts analysed. Results from experiments indicate that this additional information does not produce any advantage, i.e. the NAT comparison (NATC) and ATC yield the same results. However, the question is not whether NATC performs better than ATC. From a general point of view, in fact, the ATC method is of much greater interest than the NATC one. In fact, while in linguistic related problems the two methods perform equally, in many cases NATC are impossible to construct because outside linguistics there is no precise definition of 'word'. On the other hand the fact that ATC and NATC perform at least equally well in linguistics motivated problems is good news, because one can reasonably infer that the situation will not change drastically in situations where NATC is not available any longer.

A slightly different application of our method is that of the self-consistent classification of a corpus of sequences. In this case we do not need any information about the corpus,

J. Stat. Mech. (2005) P04002

but we are interested in observing the self-organization that arises from the knowledge of a matrix of distances between pairs of elements. A good way to represent this structure can be obtained using phylogenetic algorithms to build a tree representation of the corpus considered. In this paper we have shown how the self-organized structures observed in these trees are related to the semantic attributes of the texts considered.

Finally, it is worth stressing once again the high level of versatility and generality of our method, that applies to any kind of corpora of character strings independently of the type of coding behind them: texts, symbolic dynamics of dynamical systems, time series, genetic sequences etc. These features could be potentially very important for fields where human intuition can fail: genomics, geological time series, stock market data, medical monitoring etc.

Acknowledgments

The authors are indebted to Dario Benedetto with whom part of this work was completed. The authors are grateful to Valentina Alfi, Luigi Luca Cavalli-Sforza, Mirko Degli Esposti, David Gomez, Giorgio Parisi, Luciano Pietronero, Andrea Puglisi, Angelo Vulpiani, William S Wang for very enlightening discussions.

References

[1] Shannon C E, *A mathematical theory of communication*, 1948 *Bell Syst. Tech. J.* **27** 379
Shannon C E, 1948 *Bell Syst. Tech. J.* **27** 623
[2] Zurek W H (ed), 1990 *Complexity, Entropy and Physics of Information* (Redwood City, CA: Addison-Wesley)
[3] Benedetto D, Caglioti E and Loreto V, *Language trees and zipping*, 2002 *Phys. Rev. Lett.* **88** 048702
[4] Lempel A and Ziv J, *A universal algorithm for sequential data compression*, 1977 *IEEE Trans. Inf. Theory* **23** 337
[5] Loewenstern D, Hirsh H, Yianilos P and Noordewieret M, *DNA sequence classification using compression-based induction*, 1995 *DIMACS Technical Report* 95–04
[6] Kukushkina O V, Polikarpov A A and Khmelev D V, 2000 *Probl. Pereda. Inf.* **37** 96 (in Russian)
Kukushkina O V, Polikarpov A A and Khmelev D V, *Using literal and grammatical statistics for authorship attribution*, 2001 *Probl. Inf. Transmission* **37** 172 (transl.)
[7] Juola P, *Cross-entropy and linguistic typology*, 1998 *Proc. New Methods in Language Processing (Sidney, 1998)* vol 3
[8] Teahan W J, *Text classification and segmentation using minimum cross-entropy*, 2000 *RIAO: Proc. Int. Conf. Content-based Multimedia Information Access* (Paris: C.I.D.-C.A.S.I.S) p 943
[9] Thaper N, *MS in computer science*, 2001 *Master Thesis* MIT
[10] Baronchelli A, Caglioti E, Loreto V and Pizzi E, *Dictionary based methods for information extraction*, 2004 *Physica* A **342** 294
[11] Puglisi A, Benedetto D, Caglioti E, Loreto V and Vulpiani A, *Data compression and learning in time sequences analysis*, 2003 *Physica* D **180** 92
[12] Fukuda K, Stanley H E and Nunes Amaral L A, *Heuristic segmentation of a nonstationary time series*, 2004 *Phys. Rev.* E **69** 021108
Galván P, Carpena P, Román-Roldán R, Oliver J and Stanley H E, *Analysis of symbolic sequences using the Jensen–Shannon divergence*, 2002 *Phys. Rev.* E **65** 041905
[13] Azad R K, Bernaola-Galván P, Ramaswamy R and Rao J S, *Segmentation of genomic DNA through entropic divergence: power laws and scaling*, 2002 *Phys. Rev.* E **65** 051909
[14] Mantegna R N, Buldyrev S V, Goldberger A L, Havlin S, Peng C K, Simons M and Stanley H E, *Linguistic features of non-coding DNA sequences*, 1994 *Phys. Rev. Lett.* **73** 3169
[15] Grosse I, Bernaola-Ivan P, Carpena P, Roman-Roldan R, Oliver J and Stanley H E, *Analysis of symbolic sequences using the Jensen–Shannon divergence*, 2002 *Phys. Rev.* E **65** 041905
[16] Kennel M B, *Testing time symmetry in time series using data compression dictionaries*, 2004 *Phys. Rev.* E **69** 056208

J. Stat. Mech. (2005) P04002

[17] Mertens S and Bauke H, *Entropy of pseudo-random-number generators*, 2004 *Phys. Rev.* E **69** 055702
[18] Falcioni M, Palatella L, Pigolotti S and Vulpiani A, *What properties make a chaotic system a good pseudo random number generator*, 2005 *Preprint* nlin.CD/0503035
[19] Grassberger P, *Data compression and entropy estimates by non-sequential recursive pair substitution*, 2002 http://babbage.sissa.it/abs/physics/0207023
[20] Schuermann T and Grassberger P, *Entropy estimation of symbol sequences*, 1996 *Chaos* **6** 167
[21] Quiroga R Q, Arnhold J, Lehnertz K and Grassberger P, *Kulback–Leibler and renormalized entropies: applications to electroencephalograms of epilepsy patients*, 2000 *Phys. Rev.* E **62** 8380
[22] Kopitzki K, Warnke P C, Saparin P, Kurths J and Timmer J, *Comment on 'Kullback–Leibler and renormalized entropies: applications to electroencephalograms of epilepsy patients'*, 2002 *Phys. Rev.* E **66** 043902
[23] Quiroga R Q, Arnhold J, Lehnertz K and Grassberger P, *Reply to 'Comment on 'Kullback–Leibler and renormalized entropies: Applications to electroencephalograms of epilepsy patients'*, 2002 *Phys. Rev.* E **66** 043903
[24] Khinchin A I, 1957 *Mathematical Foundations of Information Theory* (New York: Dover)
[25] Welsh D, 1989 *Codes and Cryptography* (Oxford: Clarendon)
[26] Chaitin G J, *On the length of programs for computing finite binary sequences*, 1966 *J. Assoc. Comput. Machinery* **13** 547
[27] Chaitin G J, 2002 *Information, Randomness and Incompleteness* 2nd edn (Singapore: World Scientific)
[28] Kolmogorov A N, *Three approaches to the quantitative definition of information*, 1965 *Probl. Inf. Transmission* **1** 1
[29] Solomonov R J, *A formal theory of inductive inference*, 1964 *Inf. Control* **7** 1
Solomonov R J, 1964 *Inf. Control* **7** 224
[30] Li M and Vitányi P M B, 1997 *An Introduction to Kolmogorov Complexity and its Applications* 2nd edn (Berlin: Springer)
[31] Wyner A D and Ziv J, *The sliding-window Lempel–Ziv algorithm is asymptotically optimal*, 1994 *Proc. IEEE* **82** 872
[32] Pierce J R, 1980 *Introduction to Information Theory: Symbols, Signals and Noise* 2nd edn (New York: Dover)
[33] Farach M, Noordewier M, Savari S, Shepp L, Wyner A and Ziv J, *On the entropy of DNA: algorithms and measurements based on memory and rapid convergence*, 1995 *Proc. 6th Annual ACM-SIAM Symp. on Discrete Algorithms (San Francisco, CA, 1995)* p 48
[34] Milosavljević A, *Discovering dependencies via algorithmic mutual information: a case study in DNA sequence comparisons*, 1995 *Mach. Learn.* **21** 35
[35] Wyner A D, 1995 *Shannon Lecture, Typical Sequences and all that: Entropy, Pattern Matching and Data Compression, IEEE Information Theory Society Newsletters, June 1995*
[36] Ziv J and Merhav N, *A measure of relative entropy between individual sequences with applications to universal classification*, 1993 *IEEE Trans. Inf. Theory* **39** 1280
[37] Cai H, Kulkarni S and Verdú S, 2002 *Proc. 2002 IEEE Int. Symp. on Inf. Theory (USA)* p 433
[38] Verdú S, *Fifty Years of Shannon Theory*, 1998 *IEEE Trans. Inf. Theory* **44** 2057
[39] Sinai Y G, *On the notion of entropy of a dynamical system*, 1959 *Dokl. Akad. Nauk. SSSR* **124** 768
[40] Eckmann J P and Ruelle D, *Ergodic theory of chaos and strange attractors*, 1985 *Rev. Mod. Phys.* **57** 617
[41] Benci V, Bonanno C, Galatolo S, Menconi G and Virgilio M, *Dynamical systems and computable information*, 2004 *Discrete and Continuous Dynamical Systems* B **4** 935 and references therein [cond-mat/0210654]
[42] Boffetta G, Cencini M, Falcioni M and Vulpiani A, *Predictability: a way to characterize Complexity*, 2002 *Phys. Rep.* **356** 367
[43] Lind D and Marcus B, 1995 *Symbolic Dynamic and Coding* (Cambridge: Cambridge University Press)
[44] Bell T C, Cleary J C and Witten I H, 1990 *Text Compression* (Englewood Cliffs, NJ: Prentice-Hall)
[45] Bennett C H, Li M and Ma B, *Chain letters and evolutionary histories*, 2003 *Sci. Am.* **288** 76
[46] El-Yaniv R, Fine S and Tishby N, *Agnostic clustering of Markovian sequences*, 1997 *Adv. Neural Inf. Process. Syst.* **10** 465
[47] Kontoyiannis I, Algoet P H, Suhov Yu M and Wyner A J, *Nonparametric entropy estimation for stationary processes and random fields, with applications to English text*, 1998 *IEEE Trans. Inf. Theory* **44** 1319
[48] Nevill-Manning C, *Inferring sequential structures*, 1996 *PhD Thesis* University of Waikato
[49] Grumbach S and Tahi F, *A new challenge for compression algorithms: genetic sequences*, 1994 *J. Inf. Process. Manag.* **30** 875

J. Stat. Mech. (2005) P04002

[50] Li M, Badger J, Chen X, Kwong S, Kearney P and Zhang H, *An information based sequence distance and its application to whole (mitochondria) genome distance*, 2001 *Bioinformatics* **17** 149
[51] Menconi G, *Sublinear growth of information in DNA sequences*, 2004 *Bull. Math. Biol.* at press [q-bio.GN/0402046]
[52] Li M, Chen X, Li X, Ma B and Vitanyi P M B, *The similarity metric*, 2004 *IEEE Trans. Inf. Theory.* **50** 3250 [cs.CC/0111054v3]
[53] Cilibrasi R, Vitanyi P M B and de Wolf R, *Algorithmic clustering of music based on string compression*, 2004 *Comput. Music J.* **28** 49 [cs.SD/0303025]
[54] Londei A, Loreto V and Belardinelli M O, 2003 *Proc. 5th Triannual ESCOM Conf.* p 200
[55] Cover T and Thomas J, 1991 *Elements of Information Theory* (New York: Wiley)
[56] Kullback S and Leibler R A, *On information and sufficiency*, 1951 *Ann. Math. Stat.* **22** 79
[57] Kullback S, 1959 *Information Theory and Statistics* (New York: Wiley)
[58] Bennett C H, Gàcs P, Li M, Vitanyi P M B and Zurek W, *Information distance*, 1998 *IEEE Trans. Inf. Theory* **44** 1407
[59] Cilibrasi R and Vitanyi P M B, *Clustering by compression*, 2005 *IEEE Trans. Inf. Theory* **51** [cs.CV/0312044]
[60] Kaltchenko A, *Algorithms for estimating information distance with application to bioinformatics and linguistics*, 2004 *Preprint* cs.CC/0404039
[61] Otu H H and Sayood K, *A new sequence distance measure for phylogenetic tree construction*, 2003 *Bioinformatics* **19** 2122
[62] Lempel A and Ziv J, *On the complexity of finite sequences*, 1976 *IEEE Trans. Inf. Theory* **22** 75
[63] Baronchelli A, Caglioti E, Loreto V and Puglisi A, 2005 in preparation
[64] UE web site: http://europa.eu.int
[65] liberliber homepage: http://www.liberliber.it
[66] Stock J M and Trebbi F, 2003 *J. Economic Perspectives* **17** 177
[67] *Grunberg–Van der Jagt authorship attribution problem:*, see for instance: http://pil.phys.uniroma1.it/∼loreto/nrc1.ps
[68] Cavalli-Sforza L L and Edwards A W, *Phylogenetic analysis: models and estimation procedures*, 1967 *Evolution* **32** 550
Cavalli-Sforza L L and Edwards A W, *Phylogenetic analysis: models and estimation procedures*, 1967 *Am. J. Hum. Genet.* **19** 233
[69] Farris J S, *Distance data in phylogenetic analysis*, 1981 *Advances in Cladistics* ed V A Funk and D R Brooks (New York: New York Botanical Garden) pp 3–23
[70] Felsenstein J, *Distance methods for inferring phylogenies: a justification*, 1984 *Evolution* **38** 16
[71] Fitch W M and Margoliash E, *Construction of phylogenetic trees*, 1967 *Science* **155** 279
[72] Felsenstein J, *PHYLIP–Phylogeny inference package*, 1989 *Cladistics* **5** 164
[73] Saitou N and Nei M, *The neighbor-joining method: a new method for reconstructing phylogenetic trees*, 1987 *Mol. Biol. Evol.* **4** 406
[74] UNHCHR web site: http://www.unhchr.ch/udhr/navigate/alpha.htm
[75] Ethnologue web site http://www.sil.org/ethnologue
[76] Renfrew C, McMahon A and Trask L, *Time Depth in Historical Linguistics* vol 1 and 2 (Cambridge: The McDonald Institute for Archaeological Research) p 59
[77] Swadesh M, *Salish internal relationships*, 1950 *Int. J. Am. Linguistics* **16** 157
[78] Loreto V, Mortarino C and Starostin S, *A new approach to classification tree building in historical linguistics*, 2005 *Santa Fe Institute Monograph on Evolution Human Languages (EHL) Project*

J. Stat. Mech. (2005) P04002