Similarity metrics based on *n*-gram statistics and applications to authorship attribution problems

Chiara Basile

basile@dm.unibo.it

Dipartimento di Matematica
Università di Bologna

YEP-V, EURANDOM, Eindhoven, March 9-14, 2008

1. The problem
    - Quantitative A.A.
    - The Gramsci Project

1. The problem
   - Quantitative A.A.
   - The Gramsci Project

2. Similarity metrics
   - Definitions
   - A model
   - *n*-gram distances
   - Entropic methods

3. Experiments
   - Voting
   - Open and blind tests
   - Future developments

4. Graphs
   - Motivations and definitions
   - Eulerian circuits
   - Fun

Stylometry

Stylometry = Quantitative Authorship Attribution = detection of the author of an anonymous/apocryphal text by counting some "quantities" inside the text itself and comparing these measures to those performed on texts of known attribution (*reference set*).

Stylometry

Stylometry = Quantitative Authorship Attribution = detection of the author of an anonymous/apocryphal text by counting some "quantities" inside the text itself and comparing these measures to those performed on texts of known attribution (*reference set*).

De falso credita et ementita Constantini donatione declamatio - L. Valla, 1440

Stylometry

Stylometry = Quantitative Authorship Attribution = detection of the author of an anonymous/apocryphal text by counting some "quantities" inside the text itself and comparing these measures to those performed on texts of known attribution (*reference set*).

The aim is to identify some indicator(s) of the style of an author

Stylometry

Stylometry = Quantitative Authorship Attribution = detection of the author of an anonymous/apocryphal text by counting some "quantities" inside the text itself and comparing these measures to those performed on texts of known attribution (*reference set*).

The aim is to identify some indicator(s) of the style of an author, usually by counting some linguistic, grammatical, lexical or morphological quantities.

Stylometry

Stylometry = Quantitative Authorship Attribution = detection of the author of an anonymous/apocryphal text by counting some "quantities" inside the text itself and comparing these measures to those performed on texts of known attribution (*reference set*).

The aim is to identify some indicator(s) of the style of an author, usually by counting some linguistic, grammatical, lexical or morphological quantities.

What is **style**? i.e. Which **indicators**?

Stylometry

Stylometry = Quantitative Authorship Attribution = detection of the author of an anonymous/apocryphal text by counting some “quantities” inside the text itself and comparing these measures to those performed on texts of known attribution (*reference set*).

The aim is to identify some indicator(s) of the style of an author, usually by counting some linguistic, grammatical, lexical or morphological quantities.

What is **style**? i.e. Which **indicators**?

Examples:

De Morgan, 1882, math	Mean length of words
Mosteller & Wallace, 1888, stat	Frequency of “typical” words
Yule, 1944, stat	Lexical richness
Ledger, 1989, stat	Many indicators with no syntactical meaning
Kešelj, 2003, cs	*n*-grams

Our problem

Antonio Gramsci

(Ales, 1891 - Rome, 1937)
Italian politician, intellectual and journalist, among the founders of the Italian Communist Party.

Our problem

Antonio Gramsci

(Ales, 1891 - Rome, 1937)
Italian politician, intellectual and journalist, among the founders of the Italian Communist Party.

He wrote thousands of articles for several newspapers (*L'Ordine Nuovo, Avanti!, ...*)

Our problem

Antonio Gramsci

(Ales, 1891 - Rome, 1937)
Italian politician, intellectual and journalist, among the founders of the Italian Communist Party.

He wrote thousands of articles for several newspapers (*L'Ordine Nuovo, Avanti!, ...*), most of which he left unsigned.

Our problem

Antonio Gramsci

(Ales, 1891 - Rome, 1937)
Italian politician, intellectual and journalist, among the founders of the Italian Communist Party.

He wrote thousands of articles for several newspapers (*L'Ordine Nuovo, Avanti!, ...*), most of which he left unsigned. The same did his colleagues and fellows: Amedeo Bordiga, Palmiro Togliatti, Angelo Tasca...

Our problem

Antonio Gramsci

(Ales, 1891 - Rome, 1937)
Italian politician, intellectual and journalist, among the founders of the Italian Communist Party.

He wrote thousands of articles for several newspapers (*L'Ordine Nuovo, Avanti!, ...*), most of which he left unsigned. The same did his colleagues and fellows: Amedeo Bordiga, Palmiro Togliatti, Angelo Tasca...

Gramsci Project (*Istituto Fondazione Gramsci*, Rome): recognising Gramscian articles in view of a National Edition of Gramsci's work.

Our problem

Antonio Gramsci

(Ales, 1891 - Rome, 1937)
Italian politician, intellectual and journalist, among the founders of the Italian Communist Party.

He wrote thousands of articles for several newspapers (*L'Ordine Nuovo, Avanti!, ...*), most of which he left unsigned. The same did his colleagues and fellows: Amedeo Bordiga, Palmiro Togliatti, Angelo Tasca...

Gramsci Project (*Istituto Fondazione Gramsci*, Rome): recognising Gramscian articles in view of a National Edition of Gramsci's work.

Joint work with M. Degli Esposti (Univ. of Bologna), D. Benedetto and E. Caglioti (*La Sapienza*, Rome) and M. Lana (Univ. of Western Piedmont, Vercelli).

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts...

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts... examples:

- $\mathcal{A} = \{0, 1\}$: Bernoulli, HRV and Audio files

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts... examples:

- $\mathcal{A} = \{0, 1\}$: Bernoulli, HRV and Audio files

M. Degli Esposti, C. Farinelli, M. Manca, A. Tolomelli,
A similarity measure for biological signals: new applications to HRV analysis, *JP J Biostat., vol 1, n 1, pp 53-78 (2007)*

M. Degli Esposti, C. Farinelli, G. Menconi,
Sequence distance via parsing complexity: Heartbeat signals,
Chaos, Solitons and Fractals (2007), in press

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \dots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts... examples:

- $\mathcal{A} = \{\mathtt{0}, \mathtt{1}\}$: Bernoulli, HRV and Audio files
- $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$: DNA

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts... examples:

- $\mathcal{A} = \{\mathtt{0}, \mathtt{1}\}$: Bernoulli, HRV and Audio files
- $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$: DNA
- $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, ..., \mathtt{A}, \mathtt{B}, \mathtt{C}, ..., \text{"}\mathtt{;}\text{"}, \text{"}\mathtt{!}\text{"}, \text{"}\mathtt{.}\text{"}, ..., \mathtt{0}, \mathtt{1}, \mathtt{2}, ...\}$: texts

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts... examples:

- $\mathcal{A} = \{\mathtt{0}, \mathtt{1}\}$: Bernoulli, HRV and Audio files
- $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$: DNA
- $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, ..., \mathtt{A}, \mathtt{B}, \mathtt{C}, ..., \text{";"}, \text{"!"}, \text{"."}, ..., \mathtt{0}, \mathtt{1}, \mathtt{2}, ...\}$: texts

From our point of view a text is an element of $\mathcal{A}^*$...

Texts as symbol sequences

Some definitions:

$$\mathcal{A} = \text{a finite alphabet}$$
$$\mathcal{A}^n = \{x = (x_1, \ldots, x_n) \mid x_j \in \mathcal{A}\}$$
$$\mathcal{A}^* = \bigcup_n \mathcal{A}^n$$

Not only texts... examples:

- $\mathcal{A} = \{\mathtt{0}, \mathtt{1}\}$: Bernoulli, HRV and Audio files
- $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$: DNA
- $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, ..., \mathtt{A}, \mathtt{B}, \mathtt{C}, ..., \text{";"}, \text{"!"}, \text{"."}, ..., \mathtt{0}, \mathtt{1}, \mathtt{2}, ...\}$: texts

From our point of view a text is an element of $\mathcal{A}^*$... no grammatical structure is taken into consideration.

(Pseudo) similarity distances

A distance is any function

$$d : \mathcal{A}^* \times \mathcal{A}^* \longrightarrow \mathbb{R}$$

with three properties:

symmetric: $d(x, y) = d(y, x)$

positive: $d(x, y) \geq 0$ and $d(x, y) = 0 \Leftrightarrow x = y$

triangular: $d(x, y) \leq d(x, z) + d(z, y)$

(Pseudo) similarity distances

A distance is any function

$$d : \mathcal{A}^* \times \mathcal{A}^* \longrightarrow \mathbb{R}$$

with three properties:

$$\begin{aligned} \text{symmetric:} \quad & d(x,y) = d(y,x) \\ \text{positive:} \quad & d(x,y) \geq 0 \ \text{ and } \ d(x,y) = 0 \Leftrightarrow x = y \\ \text{triangular:} \quad & d(x,y) \leq d(x,z) + d(z,y) \end{aligned}$$

We want d to be able to detect and enhance similarities between symbolic sequences,

(Pseudo) similarity distances

A distance is any function

$$d : \mathcal{A}^* \times \mathcal{A}^* \longrightarrow \mathbb{R}$$

with three properties:

symmetric: $d(x, y) = d(y, x)$

positive: $d(x, y) \geq 0$ and $d(x, y) = 0 \Leftrightarrow x = y$

triangular: $d(x, y) \leq d(x, z) + d(z, y)$

We want d to be able to detect and enhance similarities between symbolic sequences, independently of the origin of such similarities.

(Pseudo) similarity distances

A distance is any function

$$d : \mathcal{A}^* \times \mathcal{A}^* \longrightarrow \mathbb{R}$$

with three properties:

symmetric: $d(x, y) = d(y, x)$

positive: $d(x, y) \geq 0$ and $d(x, y) = 0 \Leftrightarrow x = y$

triangular: $d(x, y) \leq d(x, z) + d(z, y)$

We want d to be able to detect and enhance similarities between symbolic sequences, independently of the origin of such similarities.

Often we use pseudo-distances...

Authors as Markov sources

A very strong hypothesis:

suppose an author writes his texts choosing at each step a new character according to some transition probabilities depending only on the last n characters (the last *n-gram*) of the generated sequence.

[1]Texts downloaded from `www.gutenberg.org`

Authors as Markov sources

A very strong hypothesis:

suppose an author writes his texts choosing at each step a new character according to some transition probabilities depending only on the last n characters (the last *n-gram*) of the generated sequence.

i.e. the author is a Markov source with finite memory n.

[1]Texts downloaded from `www.gutenberg.org`

Authors as Markov sources

A very strong hypothesis:

suppose an author writes his texts choosing at each step a new character according to some transition probabilities depending only on the last n characters (the last *n-gram*) of the generated sequence.

i.e. the author is a Markov source with finite memory n.

Let's try with Charles Dickens's *Oliver Twist, David Copperfield, Great Expectations* and *A Tale of Two Cities*[1]

[1] Texts downloaded from `www.gutenberg.org`

Authors as Markov sources

A very strong hypothesis:

suppose an author writes his texts choosing at each step a new character according to some transition probabilities depending only on the last n characters (the last *n-gram*) of the generated sequence.

i.e. the author is a Markov source with finite memory n.

Let's try with Charles Dickens's *Oliver Twist, David Copperfield, Great Expectations* and *A Tale of Two Cities*[1],

a total of ~ 4.5 millions characters.

[1] Texts downloaded from `www.gutenberg.org`

Authors as Markov sources

$n = 0$ `ttkdnnc,t ou u m hvioega t,tna keseilra`

Authors as Markov sources

$n = 0$ `ttkdnnc,t ou u m hvioega t,tna keseilra`

$n = 1$ `fin my then i win blo his owe ’se a pe p`

Authors as Markov sources

$n = 0$ `ttkdnnc,t ou u m hvioega t,tna keseilra`

$n = 1$ `fin my then i win blo his owe ’se a pe p`

$n = 4$ `where as added, in recollections, may now how him going artific chap`

Authors as Markov sources

$n = 0$ `ttkdnnc,t ou u m hvioega t,tna keseilra`

$n = 1$ `fin my then i win blo his owe ’se a pe p`

$n = 4$ `where as added, in recollections, may now how him going artific chap`

$n = 10$ `by this time, estella left me stand aside, to see if she could be easier for the wash; that’s a blazing fire.`

Some Markovian methods

Markovian authors?

by this time, estella left me stand aside, to see if she could be easier for the wash; that's a blazing fire.

Some Markovian methods

Markovian authors?

`by this time, estella left me stand aside, to see if she could be easier for the wash; that's a blazing fire.`

Some (non-metric) attribution methods based on this model:

Khmelev & Tweedie, 2001: first-order Markov chains

Some Markovian methods

Markovian authors?

```
by this time, estella left me stand
aside, to see if she could be easier for
the wash; that's a blazing fire.
```

Some (non-metric) attribution methods based on this model:

Khmelev & Tweedie, 2001: first-order Markov chains

Clement & Sharp, 2003: n^{th}-order Markov chains

Some Markovian methods

Markovian authors?

```
by this time, estella left me stand
aside, to see if she could be easier for
the wash; that's a blazing fire.
```

Some (non-metric) attribution methods based on this model:

Khmelev & Tweedie, 2001: first-order Markov chains

Clement & Sharp, 2003: n^{th}-order Markov chains

where transition frequencies (computable) play the role of transition probabilities (unknown)

Some Markovian methods

Markovian authors?

`by this time, estella left me stand aside, to see if she could be easier for the wash; that's a blazing fire.`

Some (non-metric) attribution methods based on this model:

Khmelev & Tweedie, 2001: first-order Markov chains

Clement & Sharp, 2003: n^{th}-order Markov chains

where transition frequencies (computable) play the role of transition probabilities (unknown) and a text is attributed to the author which most probably generated it.

Some Markovian methods

Markovian authors?

```
by this time, estella left me stand
aside, to see if she could be easier for
the wash; that's a blazing fire.
```

Some (non-metric) attribution methods based on this model:

Khmelev & Tweedie, 2001: first-order Markov chains

Clement & Sharp, 2003: n^{th}-order Markov chains

where transition frequencies (computable) play the role of transition probabilities (unknown) and a text is attributed to the author which most probably generated it.

A simpler idea (Kešelj, 2003): comparing *n*-gram frequencies through a similarity metric...

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Example: `feel` has $D_1 = \{\texttt{f}, \texttt{e}, \texttt{l}\}$, $D_2 = \{\texttt{fe}, \texttt{ee}, \texttt{el}\}, \ldots$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \dots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \dots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Given $x, y \in \mathcal{A}^*$ their *n*-gram distance is:

$$d_n(x,y) := \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega) + f_y(\omega)}{2}} \right)^2$$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Given $x, y \in \mathcal{A}^*$ their *n*-gram distance is:

$$d_n(x,y) := \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega)+f_y(\omega)}{2}} \right)^2$$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Given $x, y \in \mathcal{A}^*$ their *n*-gram distance is:

$$d_n(x, y) := \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega) + f_y(\omega)}{2}} \right)^2$$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Given $x, y \in \mathcal{A}^*$ their n-gram distance is:

$$d_n(x, y) := \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega) + f_y(\omega)}{2}} \right)^2$$

Strong dependance on the cardinalities of $D_n(x)$ and $D_n(y)$

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Given $x, y \in \mathcal{A}^*$ their n-gram distance is:

$$d_n(x, y) := \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega) + f_y(\omega)}{2}} \right)^2$$

Strong dependance on the cardinalities of $D_n(x)$ and $D_n(y)$ $\rightarrow$ eliminated by considering only the L commonest n-grams for each text.

Kešelj's formula

Given a text $x \in \mathcal{A}^*$ and $n \geq 1$, define:

$$f_x(\alpha_1, \ldots, \alpha_n) := \frac{\#\{i \mid x_i = \alpha_1, \ldots, x_{i+n-1} = \alpha_n\}}{|x|}$$

$$D_n(x) := \{\omega \in \mathcal{A}^n \mid f_x(\omega) > 0\}$$

Given $x, y \in \mathcal{A}^*$ their n-gram distance is:

$$d_n(x, y) := \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega) + f_y(\omega)}{2}} \right)^2$$

Strong dependance on the cardinalities of $D_n(x)$ and $D_n(y)$ $\rightarrow$ eliminated by considering only the L commonest n-grams for each text.

Kešelj won the AAAC (Juola, 2004)...

Our distance

Kešelj, 2003:

$$d_n(x,y) = \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega)+f_y(\omega)}{2}} \right)^2$$

with dictionaries cut to cardinality L (usually $L \leq 5000$...).

Our distance

Kešelj, 2003:

$$d_n(x,y) = \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega)+f_y(\omega)}{2}} \right)^2$$

with dictionaries cut to cardinality L (usually $L \leq 5000$...).

Problem: for "short" texts and "large" n, most n-grams appear just once

Our distance

Kešelj, 2003:

$$d_n(x,y) = \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{\frac{f_x(\omega)+f_y(\omega)}{2}} \right)^2$$

with dictionaries cut to cardinality L (usually $L \leq 5000$...).

Problem: for "short" texts and "large" n, most n-grams appear just once $\Rightarrow$ we do not cut the dictionaries but add a factor to Kešelj's formula:

$$\boxed{d_n(x,y) := \frac{1}{|D_n(x)| + |D_n(y)|} \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{f_x(\omega) + f_y(\omega)} \right)^2}$$

Relative entropy between authors?

Kullback-Leibler divergence (or relative entropy) between two sources with probability distributions p and q:

Relative entropy between authors?

Kullback-Leibler divergence (or relative entropy) between two sources with probability distributions p and q:

$$D(q \parallel p) \quad := \quad \limsup_{N\to\infty} \frac{1}{N} \sum_{x \in \mathcal{A}^N} q(x) \log \frac{q(x)}{p(x)}$$

Relative entropy between authors?

Kullback-Leibler divergence (or relative entropy) between two sources with probability distributions p and q:

$$\begin{aligned} D(q \parallel p) &:= \limsup_{N\to\infty} \frac{1}{N} \sum_{x\in\mathcal{A}^N} q(x) \log \frac{q(x)}{p(x)} \\ &= \sum_{s\in\mathcal{A}^n} q(s) \sum_{\alpha\in\mathcal{A}} q(\alpha|s) \log \frac{q(\alpha|s)}{p(\alpha|s)} \end{aligned}$$

for two (stationary, ergodic, possibly dependent) Markov sources with finite memory not longer than n.

Relative entropy between authors?

Kullback-Leibler divergence (or relative entropy) between two sources with probability distributions p and q:

$$D(q \parallel p) := \limsup_{N\to\infty} \frac{1}{N} \sum_{x\in\mathcal{A}^N} q(x) \log \frac{q(x)}{p(x)}$$

$$= \sum_{s\in\mathcal{A}^n} q(s) \sum_{\alpha\in\mathcal{A}} q(\alpha|s) \log \frac{q(\alpha|s)}{p(\alpha|s)}$$

for two (stationary, ergodic, possibly dependent) Markov sources with finite memory not longer than n.

A second, “information theoretical” idea: using Kullback-Leibler divergence to estimate the similarities between two authors’ styles...

Relative entropy between authors?

Kullback-Leibler divergence (or relative entropy) between two sources with probability distributions p and q:

$$D(q \parallel p) := \limsup_{N\to\infty} \frac{1}{N} \sum_{x\in\mathcal{A}^N} q(x) \log \frac{q(x)}{p(x)}$$

$$= \sum_{s\in\mathcal{A}^n} q(s) \sum_{\alpha\in\mathcal{A}} q(\alpha|s) \log \frac{q(\alpha|s)}{p(\alpha|s)}$$

for two (stationary, ergodic, possibly dependent) Markov sources with finite memory not longer than n.

A second, “information theoretical” idea: using Kullback-Leibler divergence to estimate the similarities between two authors’ styles...

$\hookrightarrow$ Benedetto, Caglioti, Loreto (2002): K-L divergence approximation (?) through data compression.

▸ tell me more

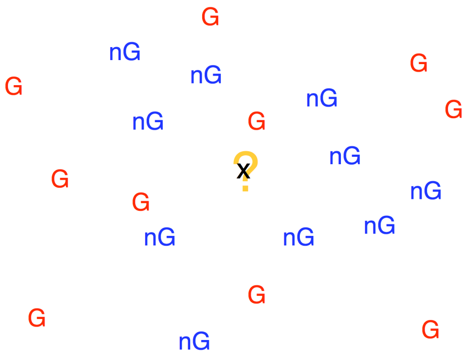
And now?

The simplest A.A. problem:
given $2N$ reference texts, N written by an author A and N by author B, how to attribute an unknown text x?

And now?

The simplest A.A. problem:
given $2N$ reference texts, N written by an author A and N by author B, how to attribute an unknown text x?

e.g. Gramscian corpus (A = Gramsci, B = "not Gramsci")

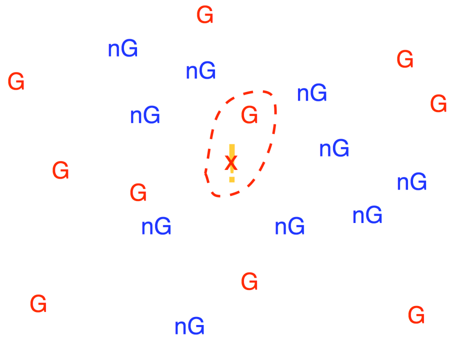


And now?

The simplest A.A. problem:
given $2N$ reference texts, N written by an author A and N by author B, how to attribute an unknown text x?

e.g. Gramscian corpus (A = Gramsci, B = "not Gramsci")

First idea: nearest neighbour

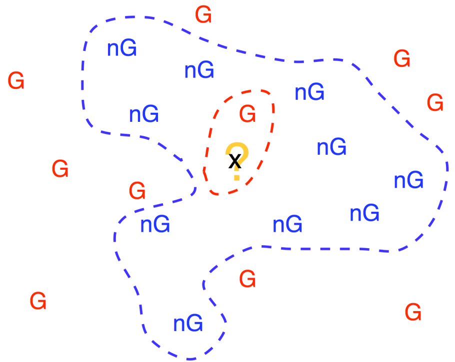


And now?

The simplest A.A. problem:
given $2N$ reference texts, N written by an author A and N by author B, how to attribute an unknown text x?

e.g. Gramscian corpus (A = Gramsci, B = “not Gramsci”)

First idea: nearest neighbour $\rightarrow$ too few information used!



And now?

The simplest A.A. problem:
given $2N$ reference texts, N written by an author A and N by author B, how to attribute an unknown text x?

e.g. Gramscian corpus (A = Gramsci, B = “not Gramsci”)

First idea: nearest neighbour $\rightarrow$ too few information used!

$\Rightarrow$ we need to take some kind of average over the available neighbours of the unknown text.

Our voting technique

And now?

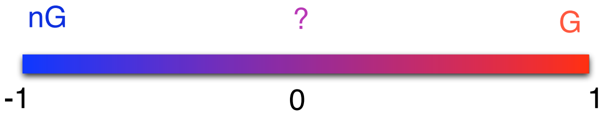
The simplest A.A. problem:
given $2N$ reference texts, N written by an author A and N by author B, how to attribute an unknown text x?

e.g. Gramscian corpus (A = Gramsci, B = “not Gramsci”)

First idea: nearest neighbour $\longrightarrow$ too few information used!

$\Rightarrow$ we need to take some kind of average over the available neighbours of the unknown text.

▸ Our voting technique



Results of a controlled test

- 50 Gramscian articles and 50 by 17 other authors

Results of a controlled test

- 50 Gramscian articles and 50 by 17 other authors
- d_{BCL} with vote for the first 3 G and nG neighbours

Results of a controlled test

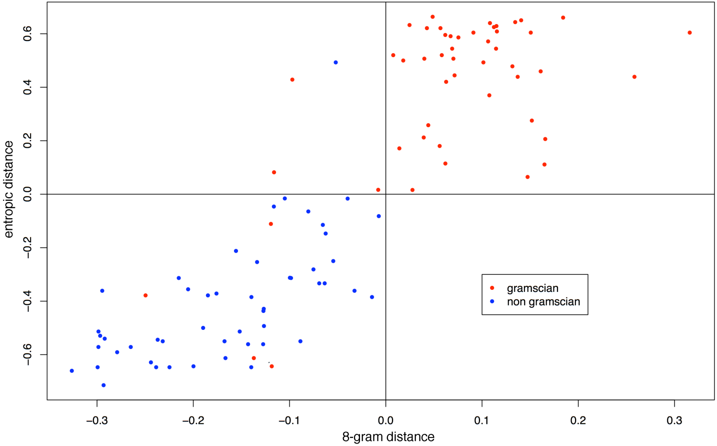
- 50 Gramscian articles and 50 by 17 other authors
- d_{BCL} with vote for the first 3 G and nG neighbours
- d_8

Results of a controlled test

- 50 Gramscian articles and 50 by 17 other authors
- d_{BCL} with vote for the first 3 G and nG neighbours
- d_8 with vote for all the 50 G and nG neighbours

Results of a controlled test

- 50 Gramscian articles and 50 by 17 other authors
- d_{BCL} with vote for the first 3 G and nG neighbours
- d_8 with vote for all the 50 G and nG neighbours



The blind test

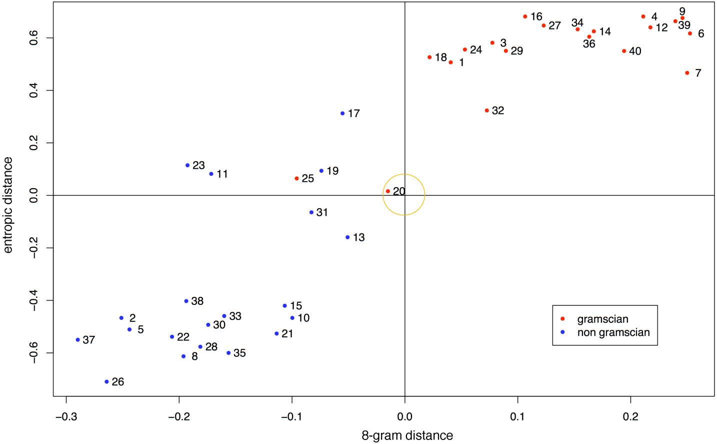
- reference set: the 100 articles of the first data set

The blind test

- reference set: the 100 articles of the first data set
- test set: 40 new articles with attribution unknown to us, but known by the *Fondazione Gramsci*

The blind test

- reference set: the 100 articles of the first data set
- test set: 40 new articles with attribution unknown to us, but known by the *Fondazione Gramsci*



Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work?

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? $\rightarrow$ many different informations extracted from texts?

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? $\rightarrow$ many different informations extracted from texts?
- why 8-grams?

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? $\rightarrow$ many different informations extracted from texts?
- why 8-grams? $\rightarrow$ low statistical significance,

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? $\rightarrow$ many different informations extracted from texts?
- why 8-grams? $\rightarrow$ low statistical significance, possible confusion with topic...

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? $\rightarrow$ many different informations extracted from texts?
- why 8-grams? $\rightarrow$ low statistical significance, possible confusion with topic...

The Project goes on

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? → many different informations extracted from texts?
- why 8-grams? → low statistical significance, possible confusion with topic...

The Project goes on:
hundreds of anonymous texts attributed (and hundreds to come)

Comments on results and doubts

Very good results: for the blind test, d_8 gave 90% Gramscian and 100% non-Gramscian articles correctly attributed.

Some open questions about *n*-grams:

- why does it work? → many different informations extracted from texts?
- why 8-grams? → low statistical significance, possible confusion with topic...

The Project goes on:
hundreds of anonymous texts attributed (and hundreds to come) and a good feedback from experts of Gramscian work at the *Fondazione*.

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

$$\texttt{item} \quad \text{and} \quad \texttt{emit} \quad \text{are the same string}$$

$\Rightarrow$ circular *n*-grams: $D_2(\texttt{item}) = \{\texttt{it}, \texttt{te}, \texttt{em}, \texttt{mi}\}$.

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

$\Rightarrow$ circular *n*-grams: $D_2(\mathtt{item}) = \{\mathtt{it}, \mathtt{te}, \mathtt{em}, \mathtt{mi}\}$.

d_n is a pseudo-distance:

- not triangular (but experimentally $D(x, y) \leq D(x, z) + D(z, y)$ for $\mathcal{A}$ large enough)

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

$\Rightarrow$ circular *n*-grams: $D_2(\texttt{item}) = \{\texttt{it}, \texttt{te}, \texttt{em}, \texttt{mi}\}$.

d_n is a pseudo-distance:

- not triangular (but experimentally $D(x,y) \leq D(x,z) + D(z,y)$ for $\mathcal{A}$ large enough)
- $d_n(x,y) = 0 \Leftarrow x = y$

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

$\Rightarrow$ circular *n*-grams: $D_2(\texttt{item}) = \{\texttt{it}, \texttt{te}, \texttt{em}, \texttt{mi}\}$.

d_n is a pseudo-distance:

- not triangular (but experimentally $D(x, y) \leq D(x, z) + D(z, y)$ for $\mathcal{A}$ large enough)
- $d_n(x, y) = 0 \not\Rightarrow x = y$ because $x \mapsto f_x$ is not injective...

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

$\Rightarrow$ circular n-grams: $D_2(\mathtt{item}) = \{\mathtt{it}, \mathtt{te}, \mathtt{em}, \mathtt{mi}\}$.

d_n is a pseudo-distance:

- not triangular (but experimentally $D(x, y) \leq D(x, z) + D(z, y)$ for $\mathcal{A}$ large enough)
- $d_n(x, y) = 0 \not\Rightarrow x = y$ because $x \mapsto f_x$ is not injective...

...some examples:

`tone` and `note` for $n = 1$;

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

$\Rightarrow$ circular *n*-grams: $D_2(\texttt{item}) = \{\texttt{it}, \texttt{te}, \texttt{em}, \texttt{mi}\}$.

d_n is a pseudo-distance:

- not triangular (but experimentally $D(x,y) \leq D(x,z) + D(z,y)$ for $\mathcal{A}$ large enough)
- $d_n(x,y) = 0 \not\Rightarrow x = y$ because $x \mapsto f_x$ is not injective...

...some examples:

`tone` and `note` for $n = 1$;
`reverse` and `severer` for $n = 2$;

The *pseudo* - distance d_n

$\mathcal{A}^*$ is for us a set of equivalence classes with respect to shift:

`item` and `emit` are the same string

$\Rightarrow$ circular *n*-grams: $D_2(\texttt{item}) = \{\texttt{it}, \texttt{te}, \texttt{em}, \texttt{mi}\}$.

d_n is a pseudo-distance:

- not triangular (but experimentally $D(x,y) \leq D(x,z) + D(z,y)$ for $\mathcal{A}$ large enough)
- $d_n(x,y) = 0 \not\Rightarrow x = y$ because $x \mapsto f_x$ is not injective...

...some examples:

`tone` and `note` for $n = 1$;
`reverse` and `severer` for $n = 2$;
`she said she should sit` and
`she sit said should she` for $n = 3$.

Why graphs?

Given $n \geq 1$, $x, y \in \mathcal{A}^N$ are said to be *n*-equivalent iff $f_x = f_y$, i.e. $d_n(x, y) = 0$.

Why graphs?

Given $n \geq 1$, $x, y \in \mathcal{A}^N$ are said to be *n*-equivalent iff $f_x = f_y$, i.e. $d_n(x, y) = 0$.

Problem 1: How to build *n*-equivalents of a text?

Why graphs?

Given $n \geq 1$, $x, y \in \mathcal{A}^N$ are said to be *n*-equivalent iff $f_x = f_y$, i.e. $d_n(x, y) = 0$.

Problem 1: How to build *n*-equivalents of a text?

For example, what is a 2-equivalent of

> *Everything should be made as simple as possible, but not simpler.*

like?

Why graphs?

Given $n \geq 1$, $x, y \in \mathcal{A}^N$ are said to be *n*-equivalent iff $f_x = f_y$, i.e. $d_n(x, y) = 0$.

Problem 1: How to build *n*-equivalents of a text?

For example, what is a 2-equivalent of

> *Everything should be made as simple as possible, but not simpler.*

like?

Problem 2: How to count *n*-equivalents of a text?

Why graphs?

Given $n \geq 1$, $x, y \in \mathcal{A}^N$ are said to be *n*-equivalent iff $f_x = f_y$, i.e. $d_n(x, y) = 0$.

Problem 1: How to build *n*-equivalents of a text?

For example, what is a 2-equivalent of

> *Everything should be made as simple as possible, but not simpler.*

like?

Problem 2: How to count *n*-equivalents of a text?

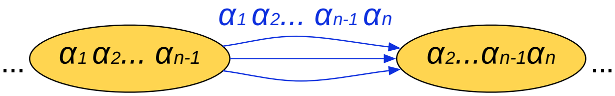
The answer comes from (multi-)graph theory.

n-gram graphs

For given $n \geq 2$ and $x \in \mathcal{A}^*$, the *n-gram graph* of x is the directed multigraph $G_n(x)$ with:

vertex set: $D_{n-1}(x)$

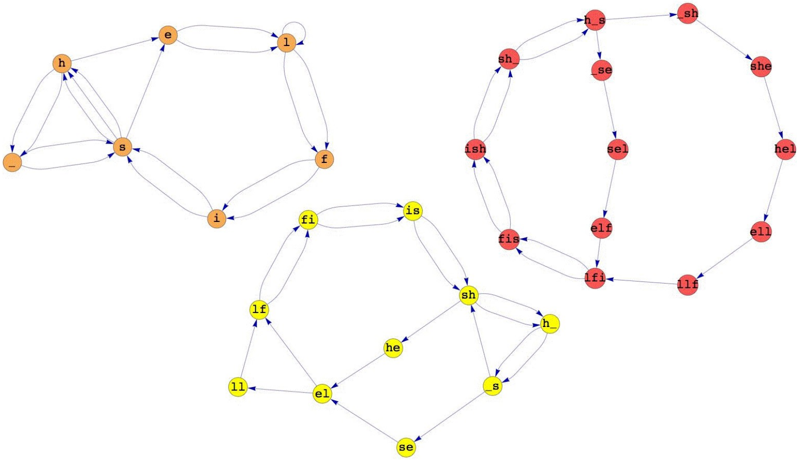
edge set: if the n-gram $(\alpha_1, \ldots, \alpha_n)$ appears m times in x, m edges link the vertex $(\alpha_1, \ldots, \alpha_{n-1})$ to the vertex $(\alpha_2, \ldots, \alpha_n)$.



$\#\{ i \mid x_i=\alpha_1 \ldots x_{i+n}=\alpha_n \} = 3$

n-gram graphs

Example: G_n(`selfish_shellfish_`) for $n = 2, 3, 4$.



Texts as Eulerian circuits

The text x can be “read” on its n-gram graph as an *Eulerian circuit*, i.e. a closed path which passes only once through each edge of the graph.

Texts as Eulerian circuits

The text x can be "read" on its n-gram graph as an *Eulerian circuit*, i.e. a closed path which passes only once through each edge of the graph.
Two n-equivalent texts share the same n-gram graph and correspond to different Eulerian circuits on it.

Texts as Eulerian circuits

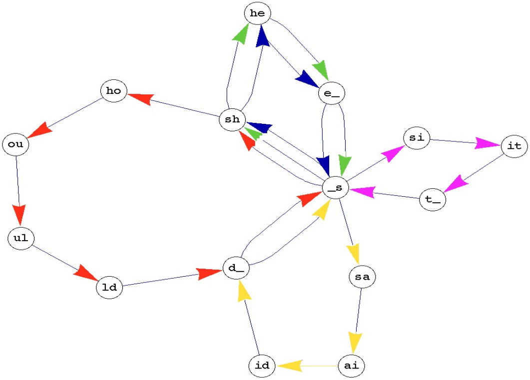
The text x can be "read" on its n-gram graph as an *Eulerian circuit*, i.e. a closed path which passes only once through each edge of the graph.
Two n-equivalent texts share the same n-gram graph and correspond to different Eulerian circuits on it.

Example:

`she said she should sit`

and

`she sit said should she`



Counting Eulerian circuits

Hence counting n-equivalents of the text x means counting Eulerian circuits on $G_n(x)$.

Counting Eulerian circuits

Hence counting n-equivalents of the text x means counting Eulerian circuits on $G_n(x)$.

BEST Theorem:
Suppose G is a directed multigraph with vertex set $V(G) = \{v_1, \ldots, v_m\}$ such that $d^+(v_i) = d^-(v_i)$ for $i = 1, \ldots, m$. Let $s(G)$ be the number of directed Eulerian circuits in G and $t_i(G)$ the number of spanning trees directed towards v_i. Then, for $i = 1, \ldots, m$,

$$\boxed{s(G) = t_i(G) \prod_{j=1}^{m} (d^+(v_j) - 1)!}$$

Counting Eulerian circuits

Hence counting n-equivalents of the text x means counting Eulerian circuits on $G_n(x)$.

BEST Theorem:
Suppose G is a directed multigraph with vertex set $V(G) = \{v_1, \ldots, v_m\}$ such that $d^+(v_i) = d^-(v_i)$ for $i = 1, \ldots, m$. Let $s(G)$ be the number of directed Eulerian circuits in G and $t_i(G)$ the number of spanning trees directed towards v_i. Then, for $i = 1, \ldots, m$,

$$\boxed{s(G) = t_i(G) \prod_{j=1}^{m} (d^+(v_j) - 1)!}$$

$d^+(v_j)$ = *out-degree* of v_j = number of edges which start in v_j.

Counting Eulerian circuits

Hence counting n-equivalents of the text x means counting Eulerian circuits on $G_n(x)$.

BEST Theorem:
Suppose G is a directed multigraph with vertex set $V(G) = \{v_1, \ldots, v_m\}$ such that $d^+(v_i) = d^-(v_i)$ for $i = 1, \ldots, m$. Let $s(G)$ be the number of directed Eulerian circuits in G and $t_i(G)$ the number of spanning trees directed towards v_i. Then, for $i = 1, \ldots, m$,

$$\boxed{s(G) = t_i(G) \prod_{j=1}^{m} (d^+(v_j) - 1)!}$$

$d^+(v_j)$ = *out-degree* of v_j = number of edges which start in v_j.
Spanning tree on G = tree which crosses all edges of G.
Spanning tree directed towards v_i = spanning tree such that $\forall\, j$ the only path linking v_j and v_i is directed towards v_i.

Counting Eulerian circuits

Now the problem is to count the spanning trees on $G_n(x)$ oriented towards any vertex v_i.

Counting Eulerian circuits

Now the problem is to count the spanning trees on $G_n(x)$ oriented towards any vertex v_i.

Theorem:
Suppose G is a directed connected non-trivial multigraph with vertex set $V(G) = \{v_1, \ldots, v_m\}$. Let $D(G)$ be the diagonal outdegree matrix of G and $A(G)$ its adjacency matrix; define $L(G) = D(G) - A(G)$ the *Laplacian matrix* of G and $L_{ii}(G)$ the minor of indexes (i, i) of $L(G)$.
Then $t_i(G) = L_{ii}(G)$.

Counting Eulerian circuits

Now the problem is to count the spanning trees on $G_n(x)$ oriented towards any vertex v_i.

Theorem:
Suppose G is a directed connected non-trivial multigraph with vertex set $V(G) = \{v_1, \ldots, v_m\}$. Let $D(G)$ be the diagonal outdegree matrix of G and $A(G)$ its adjacency matrix; define $L(G) = D(G) - A(G)$ the *Laplacian matrix* of G and $L_{ii}(G)$ the minor of indexes (i, i) of $L(G)$.
Then $t_i(G) = L_{ii}(G)$.

Remembering BEST Theorem:

$$\boxed{s(G_n(x)) = L_{kk} \prod_{j=1}^{|D_{n-1}(x)|} (d^+(v_j) - 1)!}$$

Counting Eulerian circuits

Now the problem is to count the spanning trees on $G_n(x)$ oriented towards any vertex v_i.

Theorem:
Suppose G is a directed connected non-trivial multigraph with vertex set $V(G) = \{v_1, \ldots, v_m\}$. Let $D(G)$ be the diagonal outdegree matrix of G and $A(G)$ its adjacency matrix; define $L(G) = D(G) - A(G)$ the *Laplacian matrix* of G and $L_{ii}(G)$ the minor of indexes (i, i) of $L(G)$.
Then $t_i(G) = L_{ii}(G)$.

Remembering BEST Theorem:

$$\boxed{e_n(x) \quad = \frac{1}{\prod_{i,j=1}^{|D_{n-1}(t)|} a_{ij}!} L_{kk} \prod_{j=1}^{|D_{n-1}(x)|} (d^+(v_j) - 1)!}$$

Building Eulerian circuits

Theorem (Euler):
A directed connected non-trivial multigraph G has a directed Eulerian circuit $\Leftrightarrow$ $d^+(v) = d^-(v)$ $\forall$ vertex v.

Building Eulerian circuits

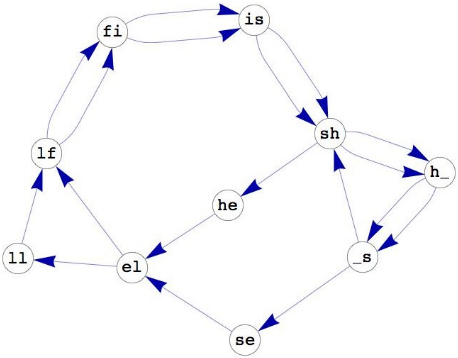
Theorem (Euler):
A directed connected non-trivial multigraph G has a directed Eulerian circuit $\Leftrightarrow d^+(v) = d^-(v)\ \forall$ vertex v.

The proof of this Theorem gives an algorithm for the construcion of Eulerian circuits.

Building Eulerian circuits

Theorem (Euler):
A directed connected non-trivial multigraph G has a directed Eulerian circuit $\Leftrightarrow$ $d^+(v) = d^-(v)$ $\forall$ vertex v.

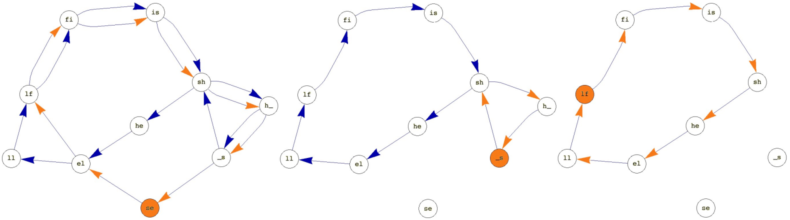
The proof of this Theorem gives an algorithm for the construcion of Eulerian circuits.



Building Eulerian circuits

Theorem (Euler):
A directed connected non-trivial multigraph G has a directed Eulerian circuit $\Leftrightarrow$ $d^+(v) = d^-(v)$ $\forall$ vertex v.

The proof of this Theorem gives an algorithm for the construcion of Eulerian circuits.



Example: `selfish_shellfish_`

Examples of *n*-equivalent texts

Everything should be made as simple as possible, but not simpler.

Examples of *n*-equivalent texts

Everything should be made as simple as possible, but not simpler.

One among its $\sim 10^{14}$ 2-equivalents:

Eve, be posing s nothoule ade but mpld ssibleryt s ashimasimpler.

Examples of *n*-equivalent texts

Everything should be made as simple as possible, but not simpler.

One among its $\sim 10^{14}$ 2-equivalents:

Eve, be posing s nothoule ade but mpld ssibleryt s ashimasimpler.

One among its 108 3-equivalents:

Everything simple, be made as possible as should but not simpler.

Examples of *n*-equivalent texts

Everything should be made as simple as possible, but not simpler.

One among its $\sim 10^{14}$ 2-equivalents:

Eve, be posing s nothoule ade but mpld ssibleryt s ashimasimpler.

One among its 108 3-equivalents:

Everything simple, be made as possible as should but not simpler.

One among its 2 4-equivalents:

Everything should be made as possible, but not simple as simpler.

A 6-equivalent of *Wish you were here*

So, so you think you can tell
heaven from hell, blue skies from pain,
can you tell a green field
from a cold steel rail?
A smile from a veil?
Do you think you can tell?

Did they get you to trade,
your heroes for ghosts?
Hot ashes for trees?
Hot air for a cool breeze?
Cold comfort for change?
And did you exchange
a walk on part in the war,
for a lead role in a cage?

How I wish, how I wish you were here,
we're just two lost soles swimming in a fish bowl,
year after year,
running over the same old ground,
but have we found the same old fears,
wish you were here.

A 6-equivalent of *Wish you were here*

So, so you were here,
we're just two lost soles swimming in a cage?

How I wish, how I wish you think you exchange
a walk on part in the same old fears,
wish you think you to trade,
your heroes for a lead role in a fish bowl,
year after year,
running over the war,
for a cold steel rail?
A smile from a veil?
Do you tell a green field
from a cool breeze?
Cold comfort for trees?
Hot air for ghosts?
Hot ashes for change?
And did you can tell?

Did the same old ground,
but have we found they get you can tell
heaven from hell, blue skies from pain,
can you were here.

Voting

1. $2N$ reference texts ordered by distance from x;

Voting

1. $2N$ reference texts ordered by distance from x;

e.g. x = *gram_27*:

1.	*gram_26*
2.	*gram_30*
3.	*gram_34*
4.	*bordiga_08*
5.	*tasca_36*
6.	*gram_49*
7.	*gram_32*
8.	*gram_46*
9.	*bordiga_09*
10.	*leonetti_27*
...	...

Voting

1. $2N$ reference texts ordered by distance from x;
2. *A*- and *B*-indexes:

$$v_A(x) = \sum_{j=1}^{N} \frac{k_A(j)}{j}, \quad v_B(x) = \sum_{j=1}^{N} \frac{k_B(j)}{j},$$

where $k_A(j)$ = position in the rank of the j-th text by A.

Voting

1. $2N$ reference texts ordered by distance from x;
2. *A*- and *B*-indexes:

$$v_A(x) = \sum_{j=1}^{N} \frac{k_A(j)}{j}, \quad v_B(x) = \sum_{j=1}^{N} \frac{k_B(j)}{j},$$

where $k_A(j)$ = position in the rank of the j-th text by A.

e.g. x = *gram_27*:

1.	*gram_26*
2.	*gram_30*
3.	*gram_34*
4.	*bordiga_08*
5.	*tasca_36*
6.	*gram_49*
7.	*gram_32*
8.	*gram_46*
9.	*bordiga_09*
10.	*leonetti_27*
...	...

$\Rightarrow$

$$v_G(gram\_27) = \tfrac{1}{1} + \tfrac{2}{2} + \tfrac{3}{3} + \tfrac{6}{4} + \tfrac{7}{5} + \dots$$

$$v_{nG}(gram\_27) = \tfrac{4}{1} + \tfrac{5}{2} + \tfrac{9}{3} + \tfrac{10}{4} + \dots$$

Voting

1. $2N$ reference texts ordered by distance from x;
2. A- and B-indexes:

$$v_A(x) = \sum_{j=1}^{N} \frac{k_A(j)}{j}, \quad v_B(x) = \sum_{j=1}^{N} \frac{k_B(j)}{j},$$

   where $k_A(j)$ = position in the rank of the j-th text by A.

3. normalized vote:

$$v(x) = \frac{v_B(x) - v_A(x)}{v_B(x) + v_A(x)};$$

   $v(x)$ is positive if $v_A(x) < v_B(x)$, negative if $v_B(x) < v_A(x)$.

Voting

1. $2N$ reference texts ordered by distance from x;
2. A- and B-indexes:

$$v_A(x) = \sum_{j=1}^{N} \frac{k_A(j)}{j}, \quad v_B(x) = \sum_{j=1}^{N} \frac{k_B(j)}{j},$$

   where $k_A(j)$ = position in the rank of the j-th text by A.

3. normalized vote:

$$v(x) = \frac{v_B(x) - v_A(x)}{v_B(x) + v_A(x)};$$

   $v(x)$ is positive if $v_A(x) < v_B(x)$, negative if $v_B(x) < v_A(x)$.

4. x is attributed to A if $v(x) > 0$, to B if $v(x) < 0$.

Voting

1. $2N$ reference texts ordered by distance from x;
2. A- and B-indexes:

$$v_A(x) = \sum_{j=1}^{N} \frac{k_A(j)}{j}, \quad v_B(x) = \sum_{j=1}^{N} \frac{k_B(j)}{j},$$

   where $k_A(j)$ = position in the rank of the j-th text by A.

3. normalized vote:

$$v(x) = \frac{v_B(x) - v_A(x)}{v_B(x) + v_A(x)};$$

   $v(x)$ is positive if $v_A(x) < v_B(x)$, negative if $v_B(x) < v_A(x)$.

4. x is attributed to A if $v(x) > 0$, to B if $v(x) < 0$. The absolute value of $v(x)$ is a maesure of the certainty of the attribution.

Voting

1. $2N$ reference texts ordered by distance from x;
2. A- and B-indexes:
$$v_A(x) = \sum_{j=1}^{N} \frac{k_A(j)}{j}, \quad v_B(x) = \sum_{j=1}^{N} \frac{k_B(j)}{j},$$
where $k_A(j)$ = position in the rank of the j-th text by A.
3. normalized vote:
$$v(x) = \frac{v_B(x) - v_A(x)}{v_B(x) + v_A(x)};$$
$v(x)$ is positive if $v_A(x) < v_B(x)$, negative if $v_B(x) < v_A(x)$.
4. x is attributed to A if $v(x) > 0$, to B if $v(x) < 0$. The absolute value of $v(x)$ is a maesure of the certainty of the attribution.

back

Approximating relative entropy

Even if authors were Markov sources with finite memory, we would not know their probability distributions and their memory length!

Approximating relative entropy

Even if authors were Markov sources with finite memory, we would not know their probability distributions and their memory length!

Approximating relative entropy

Even if authors were Markov sources with finite memory, we would not know their probability distributions and their memory length!

Approximating relative entropy

Even if authors were Markov sources with finite memory, we would not know their probability distributions and their memory length!

Merhav & Ziv, 1993:

$$D(q \parallel p) = \lim_{N \to \infty} \frac{1}{N} \left[c(y \parallel x) \log N - c(y) \log c(y) \right]$$

Approximating relative entropy

Even if authors were Markov sources with finite memory, we would not know their probability distributions and their memory length!

Merhav & Ziv, 1993:

$$D(q \parallel p) = \lim_{N\to\infty} \frac{1}{N} \left[c(y \parallel x) \log N - c(y) \log c(y) \right],$$

where:

$$\begin{aligned} x &= \text{string of length } N \text{ generated by source } p; \\ y &= \text{string of length } N \text{ generated by source } q; \end{aligned}$$

Approximating relative entropy

Even if authors were Markov sources with finite memory, we would not know their probability distributions and their memory length!

Merhav & Ziv, 1993:

$$D(q \parallel p) = \lim_{N \to \infty} \frac{1}{N} \left[c(y \parallel x) \log N - c(y) \log c(y) \right],$$

where:

$$\begin{aligned} x &= \text{string of length } N \text{ generated by source } p; \\ y &= \text{string of length } N \text{ generated by source } q; \\ c(y) &= \text{cardinality of the LZ parsing of } y \\ &\quad \text{(Lempel \& Ziv, 1976, 1977, 1978);} \\ c(y \| x) &= \text{cardinality of the LZ parsing of } y \text{ performed} \\ &\quad \text{only with substrings from } x. \end{aligned}$$

LZ78

BCL method

LZ parsing algorithm is the base of today's data compressors (zippers): *gzip, winzip, . . .*

BCL method

LZ parsing algorithm is the base of today's data compressors (zippers): *gzip, winzip, ...*

D. Benedetto, E. Caglioti, V. Loreto, 2002

BCL method

LZ parsing algorithm is the base of today's data compressors (zippers): *gzip, winzip, . . .*

D. Benedetto, E. Caglioti, V. Loreto, 2002:

given two (unknown) sources A and B, X a "long" sequence from A and y a "short" sequence from B, $X.y$ the string obtained appending y after X, L_X the length of the zipped file X and $\Delta_{Xy} := L_{X.y} - L_X$,

$$S_{AB} := \frac{\Delta_{Xy} - \Delta_{Yy}}{|y|}$$

could be an estimate of the divergence between A and B:

BCL method

LZ parsing algorithm is the base of today's data compressors (zippers): *gzip, winzip, . . .*

D. Benedetto, E. Caglioti, V. Loreto, 2002:

given two (unknown) sources A and B, X a "long" sequence from A and y a "short" sequence from B, $X.y$ the string obtained appending y after X, L_X the length of the zipped file X and $\Delta_{Xy} := L_{X.y} - L_X$,

$$S_{AB} := \frac{\Delta_{Xy} - \Delta_{Yy}}{|y|}$$

could be an estimate of the divergence between A and B: while compressing $X.y$ the zipper, being sequential, first parses X and then parses y using mostly substrings from X.

BCL distance

BCL method and applications to classification problems:

D. Benedetto, E. Caglioti, V. Loreto, Language Trees and Zipping, *Physical Review Letters 88 (2002)*

BCL distance

BCL method and applications to classification problems:

D. Benedetto, E. Caglioti, V. Loreto, Language Trees and Zipping, *Physical Review Letters 88 (2002)*

Distance function for two sources A and B:

$$\boxed{d_{BCL}(A,B) = \frac{\Delta_{Xy} - \Delta_{Yy}}{\Delta_{Yy}} + \frac{\Delta_{Yx} - \Delta_{Xx}}{\Delta_{Xx}}}$$

BCL distance

BCL method and applications to classification problems:

D. Benedetto, E. Caglioti, V. Loreto, Language Trees and Zipping, *Physical Review Letters 88 (2002)*

Distance function for two sources A and B:

$$\boxed{d_{BCL}(A, B) = \frac{\Delta_{Xy} - \Delta_{Yy}}{\Delta_{Yy}} + \frac{\Delta_{Yx} - \Delta_{Xx}}{\Delta_{Xx}}}$$

A similar distance is used for the Gramscian corpus.

back

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \mathtt{01111000110}$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{ \texttt{0},$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \mathtt{0\,1\,1\,1\,1\,0\,0\,0\,1\,1\,0}$$

$$c(y) = |\{\ \mathtt{0},\ \mathtt{1},$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{\ \texttt{0},\ \texttt{1},\ \texttt{11},$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{0 1 1 1 1 0 0 0 1 1 0}$$

$$c(y) = |\{\ \texttt{0},\ \texttt{1},\ \texttt{11},\ \texttt{10},$$

Similarity Metrics for A.A.

Chiara Basile

Appendix

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \mathtt{01111000110}$$

$$c(y) = |\{\ \mathtt{0},\ \mathtt{1},\ \mathtt{11},\ \mathtt{10},\ \mathtt{00},$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \mathtt{01111000110}$$

$$c(y) = |\{\ \mathtt{0}, \mathtt{1}, \mathtt{11}, \mathtt{10}, \mathtt{00}, \mathtt{110}\}|$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{\ \texttt{0}, \texttt{1}, \texttt{11}, \texttt{10}, \texttt{00}, \texttt{110}\}| = 6$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{\ \texttt{0},\ \texttt{1},\ \texttt{11},\ \texttt{10},\ \texttt{00},\ \texttt{110}\}| = 6$$

Variant: parse string y with respect to another string $x \in \mathcal{A}^*$, such that each phrase is the longest prefix of y which appears as a substring in x.

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{\ \texttt{0}, \texttt{1}, \texttt{11}, \texttt{10}, \texttt{00}, \texttt{110}\}| = 6$$

Variant: parse string y with respect to another string $x \in \mathcal{A}^*$, such that each phrase is the longest prefix of y which appears as a substring in x.

$$y = \texttt{01111000110}$$
$$x = \texttt{10010100110}$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{\ \texttt{0},\ \texttt{1},\ \texttt{11},\ \texttt{10},\ \texttt{00},\ \texttt{110}\}| = 6$$

Variant: parse string y with respect to another string $x \in \mathcal{A}^*$, such that each phrase is the longest prefix of y which appears as a substring in x.

$$y = \texttt{01111000110}$$
$$x = \texttt{10010100110}$$

$$c(y \parallel x) = |\{\ \texttt{011},$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \mathtt{01111000110}$$

$$c(y) = |\{\ \mathtt{0},\ \mathtt{1},\ \mathtt{11},\ \mathtt{10},\ \mathtt{00},\ \mathtt{110}\}| = 6$$

Variant: parse string y with respect to another string $x \in \mathcal{A}^*$, such that each phrase is the longest prefix of y which appears as a substring in x.

$$y = \mathtt{01111000110}$$
$$x = \mathtt{10010100110}$$

$$c(y \parallel x) = |\{\ \mathtt{011},\ \mathtt{110},$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \texttt{01111000110}$$

$$c(y) = |\{\ \texttt{0},\ \texttt{1},\ \texttt{11},\ \texttt{10},\ \texttt{00},\ \texttt{110}\}| = 6$$

Variant: parse string y with respect to another string $x \in \mathcal{A}^*$, such that each phrase is the longest prefix of y which appears as a substring in x.

$$y = \texttt{01111000110}$$
$$x = \texttt{10010100110}$$

$$c(y \parallel x) = |\{\ \texttt{011},\ \texttt{110},\ \texttt{00110}\}|$$

LZ78 parsing

Incremental LZ parsing: parse $y \in \mathcal{A}^*$ into $c(y)$ distinct substrings (phrases) such that each phrase is the shortest string which is not a previously parsed one.

$$y = \mathtt{01111000110}$$

$$c(y) = |\{\ \mathtt{0},\ \mathtt{1},\ \mathtt{11},\ \mathtt{10},\ \mathtt{00},\ \mathtt{110}\}| = 6$$

Variant: parse string y with respect to another string $x \in \mathcal{A}^*$, such that each phrase is the longest prefix of y which appears as a substring in x.

$$y = \mathtt{01111000110}$$
$$x = \mathtt{10010100110}$$

$$c(y \parallel x) = |\{\ \mathtt{011},\ \mathtt{110},\ \mathtt{00110}\}| = 3$$

back