A plagiarism detection procedure in three steps: selection, matches and “squares”

Chiara Basile - basile@dm.unibo.it

Mathematics Department
University of Bologna, Italy

PAN‘09 Workshop, San Sebastián - Donostia, 10/09/2009

Joint work with Dario Benedetto, Emanuele Caglioti, Giampaolo Cristadoro, Mirko Degli Esposti
A group of mathematicians from the Universities of Bologna and Rome La Sapienza gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.
Introduction

Once upon a time...

03/05/09
A group of mathematicians from the Universities of Bologna and Rome *La Sapienza* gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.

The competition deadline: 07/06/09
03/05/09
A group of mathematicians from the Universities of Bologna and Rome La Sapienza gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.

The competition deadline: 07/06/09 - just one month...
03/05/09
A group of mathematicians from the Universities of Bologna and Rome *La Sapienza* gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.

The competition deadline: 07/06/09 - just one month...
...and a few documents: “just” 14,428!
Once upon a time...

03/05/09
A group of mathematicians from the Universities of Bologna and Rome La Sapienza gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.

The competition deadline: 07/06/09 - just one month...
...and a few documents: “just” 14,428!

Therefore, two imperatives:
Once upon a time...

03/05/09
A group of mathematicians from the Universities of Bologna and Rome La Sapienza gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.

The competition deadline: 07/06/09 - just one month...
...and a few documents: “just” 14,428!

Therefore, two imperatives:
1. be (not only computationally) fast
Once upon a time...

03/05/09
A group of mathematicians from the Universities of Bologna and Rome La Sapienza gets to know of the Plagiarism Competition and decides to try some preliminary experiments on the external plagiarism corpus using methods developed for different tasks, like authorship recognition and text categorization.

The competition deadline: 07/06/09 - just one month...
...and a few documents: “just” 14,428!

Therefore, two imperatives:

1. be (not only computationally) fast
2. use heuristics
Where do we come from?

Various problems of *classification and clustering of symbolic sequences* (authorship attribution, classification of biological or genetic sequences, ...)
Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

The Gramsci Project

C. Basile, D. Benedetto, E. Caglioti, M. Degli Esposti
An example of mathematical authorship attribution
Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

faced using ideas coming from Information Theory, Dynamical Systems, Statistical Mechanics...
Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

faced using ideas coming from Information Theory, Dynamical Systems, Statistical Mechanics...

and usually defining some similarity metric(s) to estimate the “distance” between couples of sequences.
Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

faced using ideas coming from Information Theory, Dynamical Systems, Statistical Mechanics...

and usually defining some similarity metric(s) to estimate the “distance” between couples of sequences.

Given two texts $x$, $y$ their $n$-gram distance is:

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

where:

- $f_x(\omega) =$ frequency of the (character) $n$–gram $\omega$ in $x$;
- $D_n(x) =$ set of all the $n$–grams with non-zero frequency in $x$. 
Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

faced using ideas coming from Information Theory, Dynamical Systems, Statistical Mechanics...

and usually defining some similarity metric(s) to estimate the “distance” between couples of sequences.

Given two texts $x$, $y$ their $n$-gram distance is:

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

where:

- $f_x(\omega) =$ frequency of the (character) $n$-gram $\omega$ in $x$;
- $D_n(x) =$ set of all the $n$-grams with non-zero frequency in $x$. 

Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

faced using ideas coming from Information Theory, Dynamical Systems, Statistical Mechanics...

and usually defining some similarity metric(s) to estimate the “distance” between couples of sequences.

Given two texts $x, y$ their $n$-gram distance is:

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

where:

- $f_x(\omega) =$ frequency of the (character) $n$-gram $\omega$ in $x$;
- $D_n(x) =$ set of all the $n$-grams with non-zero frequency in $x$. 
Where do we come from?

Various problems of classification and clustering of symbolic sequences (authorship attribution, classification of biological or genetic sequences, ...)

faced using ideas coming from Information Theory, Dynamical Systems, Statistical Mechanics...

and usually defining some similarity metric(s) to estimate the “distance” between couples of sequences.

Given two texts $x, y$ their $n$-gram distance is:

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

where:

- $f_x(\omega)$ = frequency of the (character) $n$-gram $\omega$ in $x$;
- $D_n(x)$ = set of all the $n$-grams with non-zero frequency in $x$. 
Corpus statistics

The figure shows the percentage of texts, measured on a logarithmic scale, as a function of text length in characters. The graph includes four lines:

- Red line: source texts (training)
- Orange line: source texts (competition)
- Blue line: suspicious texts (training)
- Purple line: suspicious texts (competition)

The x-axis represents the text length in characters, while the y-axis shows the percentage of texts on a logarithmic scale. The data points are distributed across different ranges of text lengths, illustrating the frequency distribution of text lengths in the corpus.
First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.
1 - Selection

First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the \( n \)-gram distance for this task?
1 - Selection

First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the $n$–gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet + it takes too long
1 - Selection

First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the $n$–gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet + it takes too long ⇒ reduce the alphabet!
**1 - Selection**

First of all: reduce the search space by **selecting** a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the \( n \)-gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet + it takes too long \( \Rightarrow \) reduce the alphabet!

We converted all texts into **word lengths** (up to 9):
1 - Selection

First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the $n$-gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet + it takes too long ⇒ reduce the alphabet!

We converted all texts into word lengths (up to 9):

To be or not to be: that is the question
1 - Selection

First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the \( n \)-gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet \( + \) it takes too long \( \Rightarrow \) reduce the alphabet!

We converted all texts into word lengths (up to 9):

\[
\text{To be or not to be: that is the question} \quad \rightarrow \quad 2223224238
\]
1 - Selection

First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the \( n \)-gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet + it takes too long \( \Rightarrow \) reduce the alphabet!

We converted all texts into word lengths (up to 9):

\[
\text{To be or not to be: that is the question} \rightarrow 2223224238
\]

The value \( n = 8 \) was chosen as a compromise between

\[ \rightarrow \text{acceptable computational time (2.3 days for the whole corpus)} \]
First of all: reduce the search space by selecting a small number of suitable candidates for plagiarism for each plagiarized text.

Can we use the \( n \)-gram distance for this task?

Maybe, but there is not enough statistics using the “normal” alphabet + it takes too long \( \Rightarrow \) reduce the alphabet!

We converted all texts into word lengths (up to 9):

\[
\text{To be or not to be: that is the question} \quad \rightarrow \quad 2223224238
\]

The value \( n = 8 \) was chosen as a compromise between

- acceptable computational time (2.3 days for the whole corpus)
- a good recall (81\% of the plagiarized characters come from the first 10 neighbours \( \rightarrow \) very good! 13\% of \textit{translated} plagiarism...)
2 - Matches

Now we can perform a detailed analysis on the 7214 x 10 couples of texts, looking for common subsequences (matches) longer than a fixed threshold (e.g. 15 characters).
2 - Matches

Now we can perform a detailed analysis on the 7214 x 10 couples of texts, looking for common subsequences (matches) longer than a fixed threshold (e.g. 15 characters).

A new conversion: T9 encoding.
2 - Matches

Now we can perform a detailed analysis on the 7214 x 10 couples of texts, looking for common subsequences (matches) longer then a fixed threshold (e.g. 15 characters).

A new conversion: T9 encoding.

Why T9?

- “almost unique” translation for long enough sequences (10-15 characters);
2 - Matches

Now we can perform a detailed analysis on the 7214 x 10 couples of texts, looking for common subsequences (matches) longer than a fixed threshold (e.g. 15 characters).

A new conversion: **T9 encoding**.

Why T9?

- “almost unique” translation for long enough sequences (10-15 characters);
- it reduces the alphabet to 10 symbols ⇒ speeds up the indexing phase of the matching algorithm.
Our method

2 - Matches

Now we can perform a detailed analysis on the 7214 x 10 couples of texts, looking for common subsequences (matches) longer than a fixed threshold (e.g. 15 characters).

A new conversion: T9 encoding.

Why T9?

- “almost unique” translation for long enough sequences (10-15 characters);
- it reduces the alphabet to 10 symbols ⇒ speeds up the indexing phase of the matching algorithm.

Computation times for the whole corpus: 40 hours.
2 - Matches (continued)

suspicious-document00814.txt vs. source-document04005.txt
2 - Matches (continued)

Our method
3-“Squares”

How to identify the “squares” which are so evident in this picture?

suspicious-document00814.txt vs. source-document04005.txt
3-“Squares”

How to identify the “squares” which are so evident in this picture?

We need scalability!
3-“Squares”

How to identify the “squares” which are so evident in this picture?

We need **scalability**!

Join two matches if the following conditions hold simultaneously:

1. matches are subsequent in the suspicious file
3-“Squares”

How to identify the “squares” which are so evident in this picture?

We need scalability!

Join two matches if the following conditions hold simultaneously:

1. matches are subsequent in the suspicious file
2. matches are not superimposed in the suspicious file and their distance in the suspicious file is not larger than the length of the longest of the two sequences, scaled by $\delta_x$
3-“Squares”

How to identify the “squares” which are so evident in this picture?

Join two matches if the following conditions hold simultaneously:

1. matches are subsequent in the suspicious file
2. matches are not superimposed in the suspicious file and their distance in the suspicious file is not larger than the length of the longest of the two sequences, scaled by $\delta_x$
3. the same as 2 (with possibly a different $\delta_y$) in the source file

We need scalability!

Chiara Basile (University of Bologna)
3-“Squares”

How to identify the “squares” which are so evident in this picture?

Join two matches if the following conditions hold simultaneously:

1. matches are subsequent in the suspicious file
2. matches are not superimposed in the suspicious file and their distance in the suspicious file is not larger than the length of the longest of the two sequences, scaled by $\delta_x$
3. the same as 2 (with possibly a different $\delta_y$) in the source file

Then: repeatedly merge superimposed segments

We need scalability!
Our method

3-“Squares”

How to identify the “squares” which are so evident in this picture?

We need scalability!

Join two matches if the following conditions hold simultaneously:

1. matches are subsequent in the suspicious file
2. matches are not superimposed in the suspicious file and their distance in the suspicious file is not larger than the length of the longest of the two sequences, scaled by $\delta_x$
3. the same as 2 (with possibly a different $\delta_y$) in the source file

Then: repeatedly merge superimposed segments + run the algorithm above again with smaller parameters $\delta'_x$ and $\delta'_y$. 
3-“Squares”

How to identify the “squares” which are so evident in this picture?

`suspicious-document00814.txt vs. source-document04005.txt`
3-"Squares"

How to identify the “squares” which are so evident in this picture?

- Chiara Basile (University of Bologna)
- Plagiarism detection in three steps
- San Sebastián, 10/09/2009
Summary of the procedure

1 - Selection

2 - Matches

3 - “Squares”
Summary of the procedure

1 - Selection

The Constance letters of Charles Chapin, edited by Eleanor Early and Constance... → 397276627539...
Summary of the procedure

1 - Selection

The Constance letters of Charles Chapin, edited by Eleanor Early and Constance...

\[ \text{by the 8-gram distance} \]

\[
\begin{align*}
1) & \text{source-document04005} \\
2) & \text{source-document04080} \\
3) & \text{source-document02123} \\
4) & \text{source-document02648} \\
5) & \text{source-document03464} \\
6) & \text{source-document02737} \\
7) & \text{source-document03876} \\
8) & \text{source-document05012} \\
9) & \text{source-document04456} \\
10) & \text{source-document04223}
\end{align*}
\]

\[
\text{suspicious-document00814}
\]
Summary of the procedure

1 - Selection

2 - Matches

The Constance letters of Charles Chapin, edited by Eleanor Early and Constance...
Summary of the procedure

1 - Selection

2 - Matches

The Constance letters of Charles Chapin, edited by Eleanor Early and Constance...
Summary of the procedure

1 - Selection
2 - Matches
3 - “Squares”

suspicious-document00814.txt vs. source-document04005.txt

1496 matches
Our method

Summary of the procedure

1 - Selection
2 - Matches
3 - “Squares”

1496 matches → 244 pieces
Our method

Summary of the procedure

1 - Selection
2 - Matches
3 - “Squares”

1496 matches → 244 pieces → 16 passages

Chiara Basile (University of Bologna)

Plagiarism detection in three steps

San Sebastián, 10/09/2009
Our method

Summary of the procedure

1 - Selection
2 - Matches
3 - “Squares”

1496 matches → 244 pieces → 16 passages → 8 suspected plagiarisms

Chiara Basile (University of Bologna)
Plagiarism detection in three steps
San Sebastián, 10/09/2009
Summary of the procedure

1 - Selection
2 - Matches
3 - “Squares”

Comparison with the associated xml file...  
ok!
Conclusions

Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3$, $\delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041
Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3$, $\delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.
Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3$, $\delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.

Many possible improvements:
Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3$, $\delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.

Many possible improvements:
- less heuristics in the tuning of $\delta_x$, $\delta_y$, $\delta'_x$, $\delta'_y$... density of matches?
Results and conclusions

Results on the competition corpus, with \( \delta_x = \delta_y = 3, \delta'_x = \delta'_y = 0.5 \):

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.

Many possible improvements:

- less heuristics in the tuning of \( \delta_x, \delta_y, \delta'_x, \delta'_y \)... density of matches? Maybe they can be used to control precision, recall and granularity according to the task...
Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3, \delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.

Many possible improvements:

- less heuristics in the tuning of $\delta_x$, $\delta_y$, $\delta'_x$, $\delta'_y$ ... density of matches? Maybe they can be used to control precision, recall and granularity according to the task...
- there are certainly better ideas for the selection phase...
Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3$, $\delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.

Many possible improvements:

- less heuristics in the tuning of $\delta_x$, $\delta_y$, $\delta'_x$, $\delta'_y$... density of matches?
  Maybe they can be used to control precision, recall and granularity according to the task...
- there are certainly better ideas for the selection phase...
- try other/standard clustering algorithms
Results and conclusions

Results on the competition corpus, with $\delta_x = \delta_y = 3, \delta'_x = \delta'_y = 0.5$:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.0745
- Overall score: 0.6041

i.e. the third overall score after 0.6093 and 0.6957 of the first two.

Many possible improvements:

- less heuristics in the tuning of $\delta_x, \delta_y, \delta'_x, \delta'_y$... density of matches? Maybe they can be used to control precision, recall and granularity according to the task...
- there are certainly better ideas for the selection phase...
- try other/standard clustering algorithms

And... what about the internal plagiarism problem?
To conclude

Thank you!
Our matching algorithm

Phase 1: every source document $s$ of length $N$ is indexed (once and for all) by two vectors:
Our matching algorithm

Phase 1: every source document $s$ of length $N$ is indexed (once and for all) by two vectors:

- **index** has length $N$ and its $i^{th}$ element is the index of the previous occurrence in $s$ of the 7-gram $s_i, \ldots, s_{i+6}$
Our matching algorithm

Phase 1: every source document $s$ of length $N$ is indexed (once and for all) by two vectors:

- **index** has length $N$ and its $i^{th}$ element is the index of the previous occurrence in $s$ of the 7-gram $s_i, \ldots, s_{i+6}$
- **last** has length $10^7$ and its $j^{th}$ element is the index of the last occurrence of the 7-gram $j$ (padded with zeroes on the left, if needed) in $s$
Our matching algorithm

Phase 1: every source document \( s \) of length \( N \) is indexed (once and for all) by two vectors:

- **index** has length \( N \) and its \( i^{th} \) element is the index of the previous occurrence in \( s \) of the 7-gram \( s_i, \ldots, s_{i+6} \)

- **last** has length \( 10^7 \) and its \( j^{th} \) element is the index of the last occurrence of the 7-gram \( j \) (padded with zeroes on the left, if needed) in \( s \)

N.B. The minimum length for detected matches is 7
Appendix

Our matching algorithm

Phase 1: every source document \( s \) of length \( N \) is indexed (once and for all) by two vectors:
- **index** has length \( N \) and its \( i^{th} \) element is the index of the previous occurrence in \( s \) of the 7-gram \( s_i, \ldots, s_{i+6} \)
- **last** has length \( 10^7 \) and its \( j^{th} \) element is the index of the last occurrence of the 7-gram \( j \) (padded with zeroes on the left, if needed) in \( s \)

N.B. The minimum length for detected matches is 7

Phase 2: every suspicious document \( t \) (length \( M \)) is ran through once and for each \( k = 0, \ldots, M - 1 \) the indexes \( p = \text{last}(t_k, \ldots, t_{k+6}) \) and \( \text{index}(p) \) are used to retrieve the position of the possible matches in \( s \) without running through it again.
Our matching algorithm

Phase 1: every source document $s$ of length $N$ is indexed (once and for all) by two vectors:

- **index** has length $N$ and its $i^{th}$ element is the index of the previous occurrence in $s$ of the 7-gram $s_i, \ldots, s_{i+6}$
- **last** has length $10^7$ and its $j^{th}$ element is the index of the last occurrence of the 7-gram $j$ (padded with zeroes on the left, if needed) in $s$

N.B. The minimum length for detected matches is 7

Phase 2: every suspicious document $t$ (length $M$) is ran through once and for each $k = 0, \ldots, M - 1$ the indexes $p = \text{last}(t_k, \ldots, t_{k+6})$ and $\text{index}(p)$ are used to retrieve the position of the possible matches in $s$ without running through it again.

Total cost: $M + N$ for each couple suspicious-source.