MASTER IN
COMPUTER
SCIENCE

Authorship Clustering

Rank List Fusion using multiple Text Representations Strategies for Authorship Clustering

A Thesis Presented at the
University of Neuchâtel for the
Master of Science in
Computer Science
Degree

Raphaël Margueron
16-858-839

September, 2021

UNIVERSITÄT
BERN

UNIVERSITÉ DE
NEUCHÂTEL

UNI
FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Acknowledgements

Firstly, I want to thank my family who supported me from my firsts years at the University until today where I'm handing over my master thesis. I particularly wanted to thank Prof. Jacques Savoy, my superviser for this thesis, who not only help me throughout the study but also gave me really helpful advices and documents. My thesis could not have been completed without the University of Neuchâtel's (UniNE) computational linguistics research group and the PAN @ CLEF committee, who gave me access to corpora used in the study. I also want to thank the pan.webis.de website, which gave me free access to most of the papers studied in this thesis.

Abstract

This study aim to select different methods, text representations and distance metrics to implement and evaluate different strategies to solve the authorship clustering problem. In this case, having a set of n texts written by an unknown number of authors, the task is to regroup under the same cluster all texts written by the same author. The number of different author denoted by k can variate from 1 (all texts are written by the same author) to n (each text is written by a distinct author).

To represent the underlying style of each document various text representation have been suggested: words frequencies, lemmas frequencies, letters n-grams frequencies or part-of-speech (POS) sequences frequencies. Each text is then represented by a vector with a number of dimension corresponding to the number of features. L^1, L^2 and inner product based distances have been proposed in conjunction to compute distance between vectors representations. In addition to the vector representation, compression techniques are also applied. The Oxquarry, Brunet and St-Jean corpora are used to evaluate the system effectiveness.

In a first part, the rank lists obtained by each individual representation are used to solve the authorship clustering task. From a rank list, an automatic clustering algorithm can generate the corresponding clusters, hopefully regrouping all the texts written by a given author under the same cluster. The clustering task is achieved by using three different models based on hierarchical clustering. A Silhouette score based model, a distribution-based model and a regression-based clustering models are proposed and evaluated. The clustering obtained by these models are close to the best achievable clustering for the rank lists used. In a second part, a new rank list is created by fusing rank lists obtained using different strategies. Two approaches are explored for the fusion: one use Z-Score normalization and another one use logistic regression. The rank list obtained by fusion is shown to have better performances than the best individual rank list. In addition to the fusion, two veto strategy are proposed to try to enhance the fusion quality. The veto does not show any significant improvements with the rank lists used. Lastly, we showed that using rank lists obtained by fusion yield better clustering results than the ones obtained by individual methods.

Keywords— Authorship Clustering, Rank List, Text Representation, Silhouette, Regression, Beta distribution, Fusion, Z-Score

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In a world where the information starts to propagate fast through social networks, forums or blogs and allow anonymous communication or pseudonymous communications, security and authentication problems can occur [4] [12].

When internet crimes take place, countries security organizations work together with ISPs and websites to trace criminals using network clues. But technologies, such as: onion routing and proxies, can even provide to attackers an anonymity on a network level. Due to the increase of criminal using anonymity network-based system, new algorithm able to detect authors solely based on the text are needed [4] [13].

The field of stylometry consists in analyzing the style of a written text. Stylometric techniques can be used to identify the authorship or to draw an author profile. They can be used by journalists, law courts, cybersecurity and forensic investigators to give authorship clues for their work when dealing with completely anonymized texts, e.g., ransom notes, email threats, blog posts or cybercriminals source code comments [27]. In the current study, fake news spreading or phishing by e-mails attempts can also be detected using stylometry [22].

Developing automatic and unsupervised authorship techniques is required to solve these problems [4]. A typical unsupervised authorship problem is to find in a large corpus of texts which ones are written by the same author. This problem is called authorship clustering. An example where the authorship clustering can be useful is to be able to group text written by anonymous writers (e.g. group of terrorists or old documents from unknown authors).

One of the main challenge in this domain is to be able to create authorship clustering models which can obtain reliable results on corpora with short texts (less than 1000 words). For this study, stylometric, compression, authorship verification and fusion techniques are applied to the authorship clustering problem.

1.1 Research Questions

The main research questions for this study are:

RQ1 What are good individual techniques to identify same author documents based on the text style?

RQ2 How effective are techniques for authorship clustering?

RQ3 Using the principle of combination of evidence and the methods found in RQ1, can a combined strategy improve the effectiveness of the solution?

RQ4 Using the authorship clustering models found in RQ2, can the results obtained by RQ3 be better than the ones obtained with RQ1?

The first research question evaluates a wide range of possible methods and then fine tune the best approaches to identify pairs of documents written by the same author. The second question aim to provide methods to achieve the clustering task. By answering the third question, we aim to provide a simple framework for combining individual methods. With the fourth question, we aim to evaluate if the clustering obtained by using a combination of results is better than each individual clustering.

1.2 Contributions

In this study, the following contributions are made to the computer science computational linguistic scientific community:

1. An evaluation of different feature extraction strategy using: tokens, lemmas, n-grams, POS sequences text representation with multiple distance metrics.
2. An evaluation of the compression-based methods for authorship attribution.
3. An evaluation of the mean Silhouette score maximization for unsupervised hierarchical clustering in the authorship use case. Additionally, an optimization for the authorship clustering is presented.
4. A proposition and evaluation of a supervised authorship clustering model based on an authorship attribution modelling using two beta distributions and hierarchical clustering.
5. A proposition and evaluation of a supervised authorship clustering model based on the logistic regression and the hierarchical clustering.
6. A methodology to create robust rank lists for authorship attribution and authorship clustering, using rank list fusion based on combination of evidence.

Even though the proposed clustering methods are optimized for the authorship clustering problem, they can also be applied to many other fields, e.g.:

- Texts categorization: regrouping texts from the same categories (e.g., classify news articles according to their topics).
- Computer vision clustering: regrouping images/videos containing the same objects.
- Scientific clustering: regrouping samples from experiments sensors.

As for the clustering methods, fusion techniques proposed in this study are also applicable to other fields, such as:

- Media search / search engines: Combining results from multiple information retrieval systems (e.g., meta-search systems such as the Trivago website[1])
- Information filtering: Combining results from multiple recommender systems.
- Machine learning: Combining multiple results from LTR models (learning to rank models).

1.3 Implementation and Evaluation Methodologies

The experiments realized in this study were written in the Python programming language (version 3.8.8) [29], using the following libraries:

- Python Standard Library 3.8.8, for the data types, functional programming, file and directory access, data compression, text processing [30]
- Matplotlib, for the plot generations (v3.3.4) [31]
- Numpy, for scientific computations (v1.20.2) [32]
- SciPy, for scientific computations (v1.4.1) [33]
- Scikit-learn, for machine learning algorithms implementations (v0.24.1) [34]
- BCubed, to compute the BCubed family metrics (v1.5) [39]
- AdjustText, for automatics text adjustments in plots (v0.7.3) [40]
- Tqdm, to have progress bars for heavy computations (v4.60.0) [41]

The methods proposed are written in Python files (standard text files, extension .py). Instead, the experiments are contained in a special file format called IPython Notebook (extension .ipynb) [38]. A Jupyter environment is required to run these files. For instance, the JupyterLab environment allows markdown annotation, the possibility to run only portions of codes and even new code in the same Python kernel. By running each code portion in the same Python kernel, this allows to keep in RAM heavy computations results. This allows the programmer to dynamically implement and optimize the methods.

The complete source code can be found on GitHub[2] (see Figure 1).

[1] Trivago, a hotels and lodgings meta-search engine: `https://www.trivago.com/`

[2] `https://github.com/TheRaphael0000/master_thesis`

Figure 1: Source code repository QR Code

1.4 Overview

An overview of the current technique used in the authorship field is presented in Chapter 2. Chapter 3 shows definitions and introduces the corpora used for the evaluations. In Chapter 4 individual methods to create rank lists are described. This chapter encompasses methods to generate rank lists, using stylometric clues and compression techniques. The methods proposed in the previous chapter are used for the clustering task in Chapter 5. Chapter 6 explain rank list fusion approaches and evaluate them. At the end of this chapter, the clustering methods are evaluated using the fused rank list to hopefully obtain the best possible clustering for each corpus. Finally, Chapter 7 concludes this study by reviewing experiments results and indicate the potential clues to continue the study.

Chapter 2

State of the Art

In this section, previous works used as a knowledge base for this study are presented. Some are from papers published in scientific journals, others are from PAN @ CLEF, a scientific community which proposes shared tasks every year since 2009 in the field of text forensics and stylometry [25].

This study focus on the authorship clustering task.

Authorship clustering is a problem in which a set of documents written by different authors is given, and the goal is to group documents into clusters which only contain document written by the same author. Authorship clustering is closely related to two other authorship topics: authorship attribution and authorship verification. Authorship attribution try to determine who wrote a document, given a collection of documents. Authorship verification aim to identify if two documents are written by the same author [5]. An authorship clustering problem can be split into a series of authorship verification problems by considering every pair of documents as a verification task. The clusters are created by grouping every positively verified pairs, these are also called true links [27].

Stylometry is a domain often used with most authorship techniques. Stylometry main focus is to identify the writing methods of an author, for example: the choice of words, the combinations of words, sentence structure, punctuation [24].

When dealing with large amount of data, compression methods can be used to compute similarities. These methods were used and evaluated in Oliveira and Justino (2013) [6] for the authorship attribution task. Compression based model had shown to be a good alternative to other traditional feature and pattern based models. This method is also sometime used as a baseline [23].

In 2014, Kocher and Savoy [16] proposed SPATIUM-L1, an unsupervised authorship verification model based on the most frequent words in texts. It uses the L1 norm to be able to discriminate, if a pair of texts is written by the same author or not. The model can also answer *do not know* when not enough evidence are available to make a decision. It obtained one of the best results when compared to the other model of the PAN @ CLEF 2014.

During the PAN @ CLEF 2016 [10], in the authorship clustering shared task, the participants were given multiple collections of documents, to identify authorship links and group documents by the same author. The documents are in three different languages and single authored. The solutions proposed by Bagnall [11] for this problem was to use a recurrent neural network to create a language model for the given documents. This model can then compute similarities between documents. On the other hand, Kocher [12] create for each text a feature vectors based on the most frequent terms in the corpus. Above a certain distance threshold between the texts, it can indicate a potential authorship link.

In Kocher and Savoy's (2018) paper [16], they aim to evaluate different text representation scheme for the authorship linking task. They compared the authors style, using feature vectors constructed on the most frequent occurrences of the following text representations: words frequencies, lemma frequencies, Part-Of-Speach (POS) tags frequencies, sequences of POS tags, as well as *n*-grams frequencies. The distance measures used to compare the vectors are: L^1 norms (Manhattan, Tanimoto), L^2 norm (Matusita, Clark), inner products (Cosine distance) and the Jeffery divergence. They find out that depending on the corpus, text representation and evaluation metric used, no clear distance measures text were giving the best result for all the text representations.

In 2018, Savoy [17] show that the pen name Elena Ferrante, a well known novels author, is certainly Domenico Starnone. To do so, they apply six authorship identification models: Delta, Labbé's distance, the nearest shrunken centroids, naïve Bayes, k-nearest neighbors and character n-grams. The corpus used contain novels from forty authors (including ones from Ferrante and Starnone). They find Starnone appearing very often first place for the models used. Additionally, a lexical analysis was able to justify their conclusion.

During the PAN @ CLEF 2020, the clustering task was not one of the shared task, but the closest related task was the authorship verification. For this competition, a large corpus containing around 276,000 document pairs and smaller with 52,000 document pairs was used. The two corpora are in the English language and their documents have around 21,000 characters [23].

For this task, Weerasinghe and Greenstadt [20] proposed to use the Manhattan distance between two features vector based on multiple stylometric clues. They created the feature vector based on the following methods: Character n-grams, POS sequences, special characters, function words frequency, number of characters, number of words, average number of character per word, word-length distribution (between one and ten), the vocabulary richness (using hapax-legomenon and dis-legomenon ratios), POS chunks and NP and VP construction. For the classification, a linear regression and neural network model was trained on respectively the small and large dataset, they obtained one of the best model of the shared task. Another method for this task proposed by Araujo, Gómez and Fuentes [19] was to use a Siamese neural network using n-grams words with a size from 1 to 3 (short sequences of words) for their feature vector. Instead, Ikae [21] for this task used Labbé similarity on the most frequent tokens from each author and the ones in the English language.

Chapter 3

Definitions and Corpora

In this chapter, some basic concepts related to authorship verification, authorship attribution and authorship clustering will be introduced as well as the different corpus used. Two types of corpus were selected for this study. Three literary corpora containing excerpts of English (1 corpus) and French (2 corpora) novels. As well as the ones used for PAN @ CLEF 2016 campaign [10], which contain reviews and articles written in English, Dutch and Greek.

3.1 Definitions

This section contains basic definitions and metrics to evaluate simple properties of a corpus.

Definition 1 - Document / Text
A document or a text X is an ordered list of token. A token is a non-empty string.
Example:

$$X = (\text{"the"}, \text{"quick"}, \text{"brown"}, \text{"fox"}, \text{"."})$$

To obtain tokens from a long string containing a non-tokenized document, a tokenizer is needed. A document correspond to a sample without feature selection.

Definition 2 - Author
An author Y is a string describing the author.
Example:

$$Y = \text{"Zola"}$$

Definition 3 - Corpus
A corpus contain two lists X and Y. X contain is a list of documents X_i of size N and Y a list of authors Y_j of size k.

$$X = (X_1, X_2, X_3, X_{\dots}, X_N)$$
$$N = |X|$$

$$Y = (Y_1, Y_2, Y_3, Y_{\dots}, Y_k)$$
$$k = |Y|$$

A corpus is also called a dataset for most data science problems.

Definition 4 - Authorship
The function f, is a surjective-only function which map every text X_i to a single author Y_j

$$Y_j = f(X_i)$$

The set of $\hat{Y}_a$ is the set of document written by Y_a.

$$\hat{Y}_a = \{X_i | f(X_i) = Y_a\}$$

$$N = \sum_j^k |\hat{Y}_j|$$

In authorship attribution, the goal is to find a function $\hat{f}$, an approximation of the f function. $\hat{f}$ is found using documents with known authors. The function $\hat{f}$ can then estimate the author of a new document.

Definition 5 - Relevant set
The relevant set R contains every different pairs of documents with the same authors. Links in this set are called *true links* in this study.

$$\begin{gathered}R = \{(X_a, X_b) \\ \mid\ (f(X_a) = f(X_b)) \wedge (X_a \neq X_b) \\ \forall (X_a, X_b)\}\end{gathered}$$

The non-relevant set $\bar{R}$ contains every *false links*, documents pair with different authors.

$$\begin{gathered}\bar{R} = \{(X_a, X_b) \\ \mid\ (f(X_a) \neq f(X_b)) \wedge (X_a \neq X_b) \\ \forall (X_a, X_b)\}\end{gathered}$$

All links are contained in L, the union of the relevant set R and non-relevant set $\bar{R}$.

$$\begin{gathered}L = \bar{R} \cup R \\ R \cap \bar{R} = \emptyset \\ |L| = \frac{N * (N-1)}{2}\end{gathered}$$

Definition 6 - r Ratio [10]
The r ratio is the ratio between the number of different authors k and the number of documents N in a given corpus.

$$r = \frac{k}{N}$$

The inverse of the r ratio is equivalent to the mean number of documents per authors.

$$\frac{1}{r} = \frac{N}{k} = \frac{1}{k} \cdot \sum_{j}^{k} |\hat{Y}_j|$$

If r is close to 0, most documents are written by different authors and there is a great density of true links. On this other hand, if r is close to 1, most of the document are written by a single authors and there are few true links.

Definition 7 - True link ratio
The true link ratio denoted tl_r is the ratio between the number of true links $|R|$ and the number of links $|L|$ in a given corpus. This ratio is an alternative to the r ratio and is correlated for corpus having the same number of documents per authors.

$$tl_r = \frac{|R|}{|L|}$$

The value range in the interval $[0, 1]$.
The lower the true link ratio is, the closer to the Singleton Cluster baseline the corpus is. The Singleton Cluster baseline considers every sample as a different label (aim to estimate: every document is from a different author).
With a large true link ratio, the corpus can be estimated with a Single Cluster baseline, which consider every sample is in the same cluster (aim to estimate: every document are written by the same author).

3.2 Literature corpora

Table 1 show corpus information and statistics on the corpora. These corpora contain large texts, with 10,000 tokens in average. These texts are of high quality (very low number of spelling errors), compared to other corpus, since they come from published books. The number of texts per author is relatively large, these corpora have a low r ratio.

The Oxquarry corpus contain 52 excerpts from English novels from nine different authors (Butler, Chesterton, Conrad, Forster, Hardy, Morris, Orczy, Tressel, Stevenson). The complete list of novel and authors can be found in Annex (Table B). The Oxquarry corpus come already tokenized, which means every word and punctuation are already separated.

The Brunet French corpus contain exactly four excerpts of novels for each of the 11 authors for a total of 44 excerpts. The complete list of novel and authors can be found in Annex (Tables C). Authors present in this corpus are: Balzac, Chateaubriand, Flaubert, Marivaux, Maupassant, Proust, Rousseau, Sand, Vernes, Voltaire, Zola. Brunet is also already tokenized in two different ways: One using the actual tokens in the text, and another one using the lemma representation.

The Saint-Jean corpus is used [28]. It contains 200 excerpts from 68 French novels written by 30 different authors during the XIX century [14]. St-Jean have a token, a lemma and a POS representations. Words orthography are corrected and standardized, e.g.: M., Mr., Monsieur, monsieur. Dates of publications of each excerpt are available for this corpus. A complete list of novel and authors can be found in Annex (Table D).

St-Jean was created in such way that it can be spliced in two parts. One part contain the first 100 texts and the other one, the 100 following, which are called respectively St-Jean Serie A and St-Jean Serie B or St-Jean A and St-Jean B, in short. Both parts approximately contain the same number of authors and the same number of documents per author. Since this corpus have more documents than the other ones, both the whole and the spliced representation are used in different scenario. When the whole corpus is used, the corpus is references as St-Jean. When only certain parts are used, each individual part is mentioned. The statistics of the two individual parts and the whole corpus is displayed in Table 1.

3.3 PAN @ CLEF 2016

During PAN @ CLEF 2016 clustering campaign, a group of corpora is given to the participants [26]. The corpora are separated in two parts: a training part where 18 clustering problems and solutions are available and a second part with 18 problems without solutions. The problems are in three languages (English, Dutch, Greek) and two genres (articles and reviews) [10]. Detailed statistics on the corpora can be found in Annex (Table A).

The r ratio is closer to 1 than 0 for most of the problems, which indicate a rather larger number of single author clusters. The baseline *Singleton Cluster* (every document are considered in a different cluster) is a challenging problem to overcome for this dataset.

The mean number of tokens, for each problem in corpora, is in the range $[142-1533]$. Compared to the literary datasets presented in Section 3.2 this corresponds to approximately 85% to 98% fewer tokens.

Additionally, the true link ratio is rather low for all problem which means that without a strong system, finding the correct true links is even harder, thus Singleton Cluster can give better results than most of the standard approaches. This problematic does not promote researchers to come up with an efficient generic solution.

3.4 Pre-processing

A pre-processing of the data is realized to prepare it for the next steps. The pre-processing is in two parts. The first part is specific to each corpus, and the second is the same for every corpus.

Oxquarry and Brunet are already tokenized such that every token are on a separated line. Additionally, Brunet have a lemma representation of the texts. This representation is contained in a separated file, each lemma is on a separated line. For these two corpora, a simple script is used to parse the file and creates a words vector for each document.

St-Jean is also already tokenized with one token per line approach. Unlike the Brunet dataset, each document have its three representation (token, lemma and POS) in the same file. The three representation are separated by a comma on each line. There are a few additional preprocessing needed for this corpus.

When the word *des* (equivalent to a plural *the* in French) is encounter, the tokenizer used to create the St-Jean files created two lines for this word since it can be lemmatized into either *de* (*some/any*) or *le* (*the*). To avoid having these words weighted twice, only the first line is kept.

St-Jean also have another specificity, it contains both the numerical representation of numbers and the textual representation. For example, the number *89* is written in St-Jean as:

```
<Nombre 89>,<>,<>
quatre,quatre,72
vingt,vingt,72
neuf,neuf,72
<Fin nombre>,<>,<>
```

The first line is a tag which contain the actual number found in the text, the three next lines are the words used to spell this number in French (*quatre*: 4, *vingt*: 20, *neuf*: 9, $4 \cdot 20 + 9 = 89$) and the last lines is a tag to escape the number sequence. Only the numerical representation is kept, in the example *89*. This type of numerical representation is also used for ordinal number,

Table 1: General information and statistics on the literary corpora

Name	**Lang.**	**Authors**	**Texts**	**r**	**True Links**	**Links**	tl_r	**Avg. #Tokens**	**Avg. Token size**
Oxquarry	EN	9	52	0.173	160	1326	0.121	11650	3.819
Brunet	FR	11	44	0.25	66	946	0.07	9778	4.013
St-Jean	FR	30	200	0.15	670	19900	0.034	11533	3.928
St-Jean A	FR	17	100	0.17	330	4950	0.067	11552	3.949
St-Jean B	FR	19	100	0.19	258	4950	0.052	11513	3.907

such as *7e* (7th):

```
<Nombre 7e>,<>,<>
septième,septième,72
<Fin nombre>,<>,<>
```

When the number is already written in full letters in the text, the parser did not tokenize it this way, only one line is created.

The first two line of each document are ignored for St-Jean since they contain metadata for the document. The following information can be found: the number of tokens in the document, the name of the collection and the document ID written is full text. Some inconsistencies on these two lines have been corrected manually. For example, in some files, only one of these lines was available.

For Oxquarry, Brunet and St-Jean, the authors of each document are contained in a single text file. Each line contains the author of the document with an ID equivalent to the line number, e.g. author on line 1 is for document 1, author on line 2 is for document 2, etc.

For the PAN16 corpus, there is no document tokenization available. Plain text files are given. A simple tokenization is realized which consider every punctuation symbols (POSIX punctuation symbols class), line breaks and spaces (POSIX spaces class) as a separator for the different tokens. Since they are written in multiple languages, no further rules are applied to tokenize more effectively the texts. The problem with this tokenization method is that it removes the punctuation symbols, which carry information.

The general preprocessing applied on every document of every corpus is to encode every text with only lower case ASCII characters. By doing so, every diacritic are removed, e.g. the word *École* (school in French) is converted to *ecole*. This can create ambiguity in French but was ignored. Here is an example of ambiguity created with this method: the words *jeune* (*young* in French) and *jeûne* (*fasting* in French). This method is not used for the PAN dataset, since the Greek alphabet is not in the ASCII character set.

3.5 Vectors Distances

To be able to compare vectors, different metrics can be used depending on the usage. Usually, two types of metrics for vector comparison can be used: Vector similarities and vector distances. The first yield a large value when the two vectors are similar, and the second a small value when they are closely related. Definitions 8 to 15 are one of the few most common $L^1/L^2/cosine$-based distances. We used the variable n to indicate the vectors size.

Definition 8 - Manhattan distance

$$dist_{Manhattan}(A, B) = \sum_{i=1}^{n} |a_i - a_i|$$

The Manhattan distance is also known as the L^1 distance measure.
Example:
Let A and B be two same size vectors.

$$\begin{aligned} A &= [-2, 3, -5, 7, -11, 13] \\ B &= [1, 2, 4, 8, 16, 32] \end{aligned}$$

The Manhattan distance is computed as follows:

$$\begin{aligned} dist_{Manhattan}(A, B) =& | - 2 - 1| + |3 - 2| \\ &+| - 5 - 4| + |7 - 8| \\ &+| - 11 - 16| + |13 - 32| \\ =& | - 3| + |1| + | - 9| \\ &+| - 1| + | - 27| + | - 19| \\ =& 60 \end{aligned}$$

Definition 9 - Tanimoto distance

$$dist_{Tanimoto}(A,B) = \frac{dist_{Manhattan}(A,B)}{\sum_{i=1}^{n} \max(a_i,b_i)}$$

Tanimoto distance is a L^1-based, it's a componentwise normalized version of the Manhattan distance.

Definition 10 - Euclidean distance

$$dist_{Euclidian}(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

The L^2 distance.

Definition 11 - Matusita distance

$$dist_{Matusita}(A,B) = dist_{Euclidian}(A',B')$$
$$\text{with } A' = \sqrt{A} \text{ and } B' = \sqrt{B}$$

An L^2-based distance, using the square root of the input vectors with the Euclidean distance.

Definition 12 - Clark distance [9]

$$dist_{Clark}(A,B) = \sqrt{\sum_{i=1}^{n}\left(\frac{|a_i - b_i|}{a_i + b_i}\right)^2}$$

An L^2-based distance.

Definition 13 - Cosine distance
The cosine distance is an inner product based distance. The inner product is defined as:

$$\langle A,B\rangle = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

To compute the Cosine distance, first the cosine similarity must be computed.

$$sim_{Cosine}(A,B) = \frac{\langle A,B\rangle}{\sqrt{\langle A,A\rangle}\sqrt{\langle B,B\rangle}}$$

The cosine similarity is ranged between -1 and 1. With 1 being exactly the same, -1 the total opposite and 0 orthogonal.

$$dist_{Cosine}(A,B) = 1 - sim_{Cosine}(A,B)$$

$$dist_{Angular\_cosine}(A,B) = \frac{cos^{-1}\left(sim_{Cosine}(A,B)\right)}{\pi}$$

These measures are based on the inner product. In this study, the only inner product based metric used is the cosine distance.

Definition 14 - Kullback-Leibler divergence

$$dist_{kld}(A \mid\mid B) = \sum_{i=1}^{n} a_i \cdot \log(\frac{a_i}{b_i})$$

Kullback-Leibler divergence use the principle of relative entropy.

Definition 15 - Jeffrey divergence [9]

$$dist_{j\_divergence}(A \mid\mid B) = \sum_{i=1}^{n}(a_i - b_i) \cdot \log(\frac{a_i}{b_i})$$

It represents the difference between two probability distribution.

3.6 Rank lists

Rank lists are used to order objects such that the most interesting object is at the top and every subsequent object become less interesting. In information retrieval systems, rank lists are used to order the results from the most relevant to the user's query to the least relevant. For the authorship verification problem, the rank list can also be used to order links.

3.6.1 Definition and Example

Definition 16 - Ranked list for authorship verification

$$L = ([(X_a, X_b) : Score(X_a, X_b)] \, | X_a \neq X_b \forall (X_a, X_b))$$

$$|L| = \frac{N \cdot (N-1)}{2}$$

A ranked list denoted L for the authorship verification problem is an ordered list containing document pairs and a score for the pair. These pairs are also called links. In most cases, the rank list contain every possible pairs of documents.

The ranked list is ordered by the score, such that the most similar document pair is at the top of the list. The least similar document pair is at the bottom of the list.

When the scoring function is based on a distance metrics, the rank list is sorted in increasing order. For the scoring function based on similarity, the rank list is sorted in decreasing order.

The most similar documents pairs are the most likely written by the same author. Thus, the top ranks should contain pairs of documents written by the same author.

The computational cost to compute a rank list is $\frac{N \cdot (N-1)}{2}$, with N the number of documents. Thus, the computation complexity is $O(N^2)$. The computation cost to sort the rank list is ignored since it has a complexity of $O(N \log(N))$. The space complexity is also $O(N^2)$, for each document pair, the pair and the score have to be stored.

Example 1 shows the creation of a rank list using two-dimensional vectors and the Manhattan distance (presented in Section 3.5).

3.6.2 Evaluation Metrics

In order to know the quality of a rank list, multiple rank list evaluation metrics are used and presented in this section. Definitions in this section are adapted versions of the ones from Kocher and Savoy [16]. The presented metrics are also well know in the authorship verification and the information retrieval field.

Example 1: Rank list computation using two-dimensional vectors and the Manhattan distance

(a) List of two-dimensional vectors

Vector ID	Vector
0	$[0, 0]$
1	$[1, 2]$
2	$[4, 6]$
3	$[1, 4]$

(b) Pairwise Manhattan distances

Vector Pair IDs	$dist_{Manhattan}(A, B)$
(0, 1)	$\|0-1\| + \|0-2\| = 3$
(0, 2)	$\|0-4\| + \|0-6\| = 10$
(0, 3)	$\|0-1\| + \|0-4\| = 5$
(1, 2)	$\|1-4\| + \|2-6\| = 7$
(1, 3)	$\|1-1\| + \|2-4\| = 2$
(2, 3)	$\|4-1\| + \|6-4\| = 5$

(c) Ordered rank list by distances

Rank	Vector Pair IDs	$dist_{Manhattan}(A, B)$
1st	(1, 3)	2
2nd	(0, 1)	3
3-4rd	(2, 3)	5
3-4th	(0, 3)	5
5th	(1, 2)	7
6th	(0, 2)	10

Definition 17 - Relevant link [16]
A relevant link is a link in the relevant set. These are also called *true links* in this study. The ones outside the relevant set are called *false links*. The relevant set contains every document pair written by the same author, see Definition 5.

$$relevant(l_i) = \begin{cases} 1, & if \ l_i \in R \\ 0, & otherwise \end{cases}$$

Definition 18 - Precision@k [16] [36]
The precision@k is a function which take a positive integer k, with k < |L|

$$precision(k) = \frac{1}{k} \sum_{j=1}^{k} relevant(j)$$

Definition 19 - High precision [16]
The high precision (HPrec) represent the maximal rank j in the rank list such that the precision is still 100%.

$$HPrec = \max\{i \in \mathbf{N} | precision(i) = 1\}$$

This metric is in the range $[0, |R|]$. $HPrec = 0$ means the first pair in the rank list is incorrect. $HPrec = |R|$ means every true links are ranked in the top part of the rank list.

Definition 20 - R-Precision [16] [36]
The R-Precision (RPrec) is the precision in the rank list at rank |R| (Precision@r). With R being the relevant set (Definition 5).

$$RPrec = precision(|R|)$$

The RPrec value is in the range $[0, 1]$. With $RPrec = 0$, every link in the first $|R|$-ranks are not in the relevant set. And $RPrec = 1$, every link in the first $|R|$-ranks are in the relevant set.

Definition 21 - Average Precision (AP) [36]
The mean over the precision@k each time a relevant link is retrieved.

$$AP = \frac{1}{|R|} \sum_{j=1}^{|L|} precision(j) \cdot relevant(j)$$

The average precision can be considered as an approximation of the area under the precision-recall curve.

The average precision, the R-Precision and the High Precision are strongly correlated. Thus, for this study, on some experiment, only the average precision is computed.

Example 2 showcases each metric for the rank list evaluation.

3.6.3 Rank Lists Relationship with Distance Matrix

When computing the rank lists, each document pair have its distance calculated. These can be represented into a matrix. In this matrix, each document represent a row and a column. The matrix elements are the distances between the two documents of the row and column. For the non-commutative distances functions, the whole matrix is used. For the commutative distances functions, only a triangle or symmetric matrix is required.

Example 2: Rank list evaluation example

(a) Documents, authorship and rank list

Suppose that a corpus contain 4 documents. Documents 0, 1 and 3 are written by the same author A. Document 2 is written by author B.
The following relevant set R and non-relevant set $\bar{R}$ can be computed using this information.

$$\begin{aligned} R =& \{(0,1),(0,3),(1,3)\} \\ \bar{R} =& \{(0,2),(1,2),(2,3)\} \\ |L| =& |R| \cup |\bar{R}| = 6 \end{aligned}$$

Suppose that these links are in a rank list with the following order:

$$((1,3),(0,1),(2,3),(0,3),(1,2),(0,2))$$

(b) Precision@k

Rank	Pair IDs	Pair $\in R$	Precision@k
1st	(1, 3)	Yes	$1/1 = 1.00$
2nd	(0, 1)	Yes	$2/2 = 1.00$
3rd	(2, 3)	No	$2/3 = 0.66$
4th	(0, 3)	Yes	$3/4 = 0.75$
5th	(1, 2)	No	$3/5 = 0.60$
6th	(0, 2)	No	$3/6 = 0.50$

(c) High precision (HPrec)

$$\begin{aligned} HPrec &= \max\{i \in \mathbf{N} | precision(i) = 1\} \\ &= \max\{1, 2\} = 2 \end{aligned}$$

(d) R-Precision (RPrec)

$$RPrec = precision(|R|) = precision(3) = 0.66$$

(e) Average Precision (AP)

$$\begin{aligned} AP &= \frac{1}{|R|} \sum_{j=1}^{|L|} precision(j) \cdot relevant(j) \\ &= \frac{1}{3} \sum_{j=1}^{6} precision(j) \cdot relevant(j) \\ &= \frac{1}{3}(1.00 \cdot 1 + 1.00 \cdot 1 + 0.66 \cdot 0 \\ &+ 0.75 \cdot 1 + 0.60 \cdot 0 + 0.50 \cdot 0) \\ &= \frac{1}{3}(1.00 + 1.00 + 0.75) = 0.92 \end{aligned}$$

Example 3: Distances matrix and Rank lists

(a) Rank list for distance matrix from Example 3b

Rank	Vector Pair IDs	Distance
1st	(1, 3)	2
2nd	(0, 1)	3
3-4rd	(2, 3)	5
3-4th	(0, 3)	5
5th	(1, 2)	7
6th	(0, 2)	10

(b) Distance matrix for rank list from Example 3a

	0	1	2	3
0	-	3	10	5
1	-	-	7	2
2	-	-	-	5
3	-	-	-	-

The transformation can be effectuated either from a rank list to a distance matrix or from a distance matrix to a rank list. For some algorithm or computations, the distance matrix representation is preferable and for others the rank list representation is better.

Example 3 show the two representations for the same distances.

3.6.4 Evaluation Comparison

To be able to compare rank list evaluation, a gain strategy is used. In this study, the gain is defined as the difference between two evaluations. It can be either positive or negative. A positive gain indicates that the rank list have better performances over another. When the gain is negative, the rank list have worse results than the other one.

The fusion methods proposed in this study transform multiple rank lists into a single rank list. Thus, having aggregation strategies is required.

When multiple rank list need to be compared to a single one, two simple aggregation methods are used. One use the mean of the results and the second the maximal value. In this study, these are called respectively Single-Mean and Single-Max. These aggregations strategies can be used in conjunction with the gain definition to compare one rank list to multiple rank lists.

Example 4: Gain

(a) Gain (One to one)

Here two rank lists are compared by evaluating the gain in average precision.

Rank list	**AP**
Rank list A	0.8
Rank list B	0.6
Gain	**AP gain**
A gain over B	$0.8 - 0.6 = +0.2$

(b) Gain (One to many)

Here one rank list is compared to two others by evaluating the gain in average precision using two aggregations strategies (Single-Mean, Single-Max).

Rank list	**AP**
Rank list A	0.8
Rank list B	0.6
Rank list C	0.9
Aggregation	**AP**
A-B (Single-Mean)	$(0.8 + 0.6)/2 = 0.7$
A-B (Single-Max)	$\max(0.8, 0.6) = 0.8$
Gain with aggregation	**AP gain**
C gain over A-B (Single-Mean)	$0.9 - 0.7 = +0.2$
C gain over A-B (Single-Max)	$0.9 - 0.8 = +0.1$

Example 4 shows the gain computation using average precision on rank lists and the aggregation with Single-Mean and Single-Max.

3.7 Clustering

3.7.1 Definition and Examples

The clustering aim to group items such that most similar items appear in the same group. These groups of similar items are called clusters.

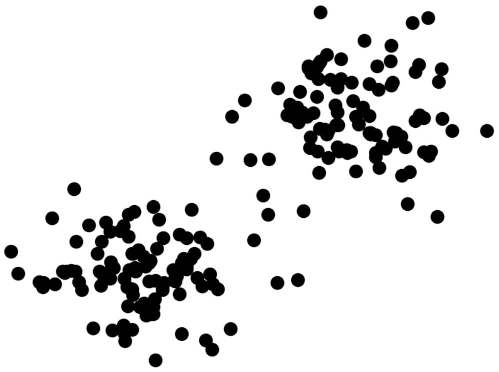
Example 5 shows the clustering task for 2D points. The points are grouped according to their distance.

For the case of authorship clustering, finding the authorship function can be useful (Definition 4). But this task can be hard since the real author (ground truth labels) are generally not available. Without example, finding the authorship function is not possible.

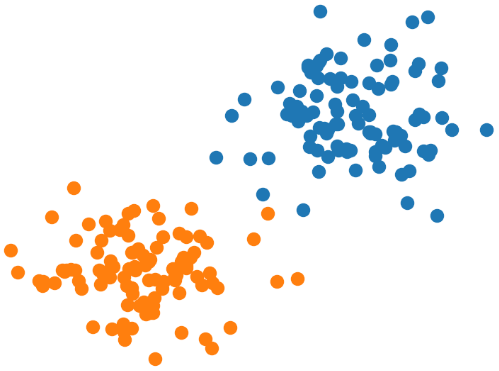
For the authorship clustering, we only aim to group similar texts instead of finding the actual authors. This task can be solved by finding the relevant set (Definition 5).

Example 5: Clustering with 2D points

(a) Unlabeled data, each point represent a sample



(b) A possible good clustering result, each color represent a cluster



To find clusters from the relevant set, a simple method is to, consider each pair of documents in the relevant set as *in the same cluster*. The clusters are created using transitivity.

One main issue arise with this method. If the relevant set is not perfect (documents pairs with actual different authors are in the relevant set), the two authors will be in the same cluster. Noisy relevant set can quickly considered every document in a single cluster with this method.

Chapter 5 will introduce and evaluate methods using rank lists to create the clusters instead of the relevant set.

3.7.2 Clustering Evaluation Metrics

The methods introduced in this section are applied to evaluate and compare clustering models. Therefore, the ground truth labels are required. The metrics used are from the BCubed family (also called B^3, in this report) [3].

BCubed has shown to satisfy the four following important constraints when evaluating clusterings [3] :

1. *Cluster Homogeneity*: different authors should be in the different clusters.
2. *Cluster Completeness*: same authors should belong to the same cluster.
3. *Rag Bag constraint*: noisy or miscellaneous authors should be in the same cluster and not in *healthy* clusters.
4. *Cluster size vs quantity constraints*: favor large clusters.

These metrics are used to evaluate the clustering task during the PAN @ CLEF competitions [10]. In addition to the BCubed family, this study introduce the *cluster difference*, a simple metric to evaluate the clustering results.

Definition 22 - Correctness [3]
The *BCubed* metric family is based on the following *Correctness* principle. Let L(e) and C(e) be the author and the cluster of an element e (an element is document in the case of authorship clustering). The Correctness is following the biconditional condition on the author and cluster equality.

$$\text{Correctness}(e, e') =$$

$$\begin{cases} 1, & \text{if}(L(e) = L(e')) \iff (C(e) = C(e')) \\ 0, & \text{otherwise} \end{cases}$$

$$\text{biconditional} : A \iff B \equiv (A \land B) \lor (\neg A \land \neg B)$$

In other terms, the correctness has a value of 1 (100% correct) if the two elements both are in the same cluster and have the same author OR are both in a different cluster and have a different author.

Definition 23 - *BCubed* Precision [3]
The *BCubed* Precision correspond to the correctness average for all elements on the average of all element such that **their clusters are the same**.

$$B^3_{precision} = \text{avg}_e[\text{avg}_{e' C(e)=C(e')}[\text{Correctness}(e, e')]]$$

Definition 24 - *BCubed* Recall [3]
The *BCubed* Recall correspond to the correctness average for all elements on the average of all element such that **their authors are the same**.

$$B^3_{recall} = \text{avg}_e[\text{avg}_{e' L(e)=L(e')}[\text{Correctness}(e, e')]]$$

Definition 25 - *BCubed* F_1 Score [3] [37]
BCubed F_1 Score uses the F_β metric with $\beta = 1$.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

The *BCubed* F_1 Score is thus computed the following way.

$$B^3_{F_1} = 2 \cdot \frac{B^3_{precision} \cdot B^3_{recall}}{B^3_{precision} + B^3_{recall}}$$

Definition 26 - Cluster difference
This metric aim to evaluate if the clustering model have found the right number of cluster. The cluster difference is the number of cluster found p minus the actual number of cluster k (number of distinct authors in the corpus).

$$Cluster_{diff} = p - k$$

A positive value indicates an overestimation of the real number of cluster, a negative value indicate the underestimation. Zero indicate that the right number of cluster was found.
This value can also be used to summarize if the B^3_{recall} is greater or not than the $B^3_{precision}$. A positive Cluster diff should indicate a larger $B^3_{precision}$ than B^3_{recall}, and vice versa.
This value can be normalized by the number of documents N, which correspond to the difference of the r ratios. This is useful when comparing corpora results with different number of clusters.

$$r_{diff} = \frac{p}{N} - \frac{k}{N} = \frac{p-k}{N}$$

As stated in the PAN16 evaluation campaign paper, estimating correctly the number of clusters and the r ratio is a first step to produce a good clustering [10].

Chapter 4

Individual Rank Lists Methods

In this chapter, the goal is to find individual consistent methods which yield a good rank list for the authorship verification task. These rank list can also be used for the authorship clustering task, which is explored in Chapter 5.

The method adopted was to compare documents using the authorship linking strategies presented in *Evaluation of text representation schemes and distance measures for authorship linking* [9]. The strategy used in this paper is based on an approach called most frequent (MF). This method aim to compare the similarity of documents by considering recurrent stylistic textual patterns. Text patterns can be, for example, from either recurrent words, short sequences of characters (known as n-grams) or part-of-speech (POS).

Another possible method is to compare similarities between document with compression techniques. This method is also slightly explored in this study.

For each method, the parameters that give the best results over the three corpora are retained and summarized at the end of this chapter.

4.1 Most Frequent Approach

The first method to represent and compare the documents is called: Most frequent (MF). This method try to compare documents by creating feature vectors containing frequencies of the most frequent items in a corpus for a specific text representation (TR). A text representation can be for example words in the text or any new representation generated from the text. This method is a generalized approach of the most frequent words technique (MFWs), which consider words as items. In the case where words are used, the most frequent items will be the most common words in a corpus.

By only considering the most frequent items, this ensures that no all the document details will be synthesized in the vector, but rather the most important in the document. The vector will thus contain the style of the document rather than the topic of the document. When comparing these vector, this method focus on the difference between the author styles rather than the topic of the documents or the time period the text was written in. This assumption is experimented in this study in Section 4.2.4.

Previous studies have shown the importance of having documents of good quality and with at least 5000 tokens to have reliable results. Skilled authors can easily change their style to imitate others for small texts, but it becomes more difficult for larger texts [24]. This problematic is explored in Section 4.2.3.

To be able to express a document as a feature vector of size n, the strategy used is to find the n most frequent (n-MF) items in a corpus. For each document, compute the item relative frequency on the n-MF items. The item occurrence vector is the number of times each item is present in a document text representation. The item relative frequency vector is obtained by normalizing the item occurrence vector such that its L_1 norm is 1, see Definition 27.

Definition 27 - L_1 norm
The L_1 norm for a vector x correspond to the sum all its elements.

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

By normalizing the vector, this ensures that long and short texts are comparable. The item relative frequency vector will represent the MF items proportions contained in a document.

Example 6: MF vector computation, example with tokens

(a) Document X

"*i realize now , that i was not looking in . i was looking out . and he , on the other side was looking in .*"[15]

(b) Vector

Suppose that the 5-MF tokens for this text's **corpus** are "*the, was, i, she, he*".
The vector using the MF strategy in Document X is thus:

	the	was	i	she	he
occurrences	1	3	3	0	1
relative frequency	1/8	3/8	3/8	0/8	1/8

For example, if a corpus contains novels of multiple genres (e.g. Comedy, Sci-fi, Fantastic, Romance) adopting the right n, most of the items specific to the genre will be discarded, since their frequency should be lower than the non topic related items which are contained in every document. The frequencies of these items should reflect the style of the author.

In this study, multiple text representation are explored, such as tokens/lemmas (Section 4.2), substrings (Section 4.3) and POS sequences (Section 4.4).

Example 6 shows the computation of the MF items vector, using words as items.

4.1.1 Normalization

Applying a normalization can improve the performances of the distance metrics presented in Section 3.5. The Z-Score normalization is one of them, see Definition 28.

In this study, the following distance metrics are always normalized using the Z-Score normalization: Manhattan, Euclidean and Cosine Distance, since they tend to produce better results when normalized. The Manhattan distance using the Z-Score normalization on MF words vectors is also called Delta model [24].

Definition 28 - Z-Score normalization [24]

$$ZScore(X) = \frac{X - \mu}{\sigma}$$

Z-Score normalize a vector X, such that the resulting vector have a 0 mean and a standard deviation of 1. When using the Z-Score normalization on MF vectors, μ and σ usually are vectors containing the mean and the standard deviation of each item in the corpus.

4.1.2 Smoothing

Relative item frequencies can be considered as a probability of occurrence based on the maximum likelihood principle. The main problem with this approach is that the frequent items have their probability overestimated and low frequency items (or the ones that does appear) have their probability underestimated. For example, if an item is not present in a document its relative item frequency is 0, but this should not mean that the probability of the author using this item is 0.

The solution proposed is to use a smoothing technique such as the Lidstone smoothing, presented in Definition 29. Smoothing techniques can help distance functions based on probabilities, such as the Kullback-Leibler Divergence [24].

In this study, except stated otherwise, the Lidstone smoothing is used with λ =0.1. Previous research show that it tends to produce better results [24].

Definition 29 - Lidstone smoothing [24]
$p(t_i, X_j)$ denotes the probability of occurrence of an item t_i in the document X_j. Using the maximum likelihood principle with the Lidstone smoothing this probability is estimated as :

$$p(t_i, X_j) = \frac{tf_{i,j} + \lambda}{|X_j| + \lambda \cdot |V|}$$

$tf_{i,j}$ is the number of occurrences of the item t_i in the document X_j. $|V|$ is the size of the vocabulary. λ is a small value ($\lambda = 1$, is a special case called Laplace smoothing). $|X_j|$ is the total number of items in the document X_j.

4.2 MF Tokens and Lemmas

This section aim to evaluate the vector representation using the most frequent (MF) tokens and lemmas.

4.2.1 MF Tokens and Lemmas Method

The MF method only have one parameter: n. n is the number of most frequent items to consider. For the tokens and lemmas text representation, no clear n value is to choose over others. Depending on the document length, previous studies have shown that using a n value between 50 and 500 tends to produce good results for the token text representation [8].

One of the main advantage of this representation is that they can be easily explained. The feature vectors can be show to the users to justify the answers. It represents the proportion of the most frequent words in a document. If n is too large to present to the user, a dimensionality reduction with, for example, the principal component analysis (PCA) can be helpful [24].

Instead of using directly the words to create the feature vector, another possibility is to use the lemma corresponding to each word, e.g. the sentence *i saw two men with a saw* its lemmatized version is: *i see two man with a saw*. This requires an advanced text preprocessing, but can help remove ambiguity. In the previous example, the lemmatization remove the ambiguity caused by *saw*. *Saw* can either be: The past tense of the verb *to see* or the *saw*, the tool. The lemmatization algorithm, using the context of the sentence, can remove the ambiguity.

The 40 most frequent tokens and lemmas in the St-Jean corpus are showed in Annex (Table E and Table F).

4.2.2 MF Tokens and Lemmas evaluation

Here, the goal is to compare two similar text representations, which are the token representation and the lemmatize representation. Rank list using these representations are created and evaluated. The corpus used for this experiment is St-Jean, since have the two text representation and is a large corpus.

The proposed methodology is to create rank lists for every proposed distance measures (ref. Section 3.5) on the two text representations. The n value is also explored between 250 and 2000 with a step of 250.

The two plots in Figure 2 shows the average precision (AP) for the resulting rank lists over n for each distance measure. Table G in Annex shows the values used to create this graph.

This representation seem to have good results with $n = 500$ for most of the distance metrics, this corroborates previous studies results. Few distance metrics, such as Manhattan distance, Tanimoto distance or Clark distance can give better results with slightly bigger vectors (750-MF or 1000-MF tokens/lemma). For most distance metrics, the token representation provide on both corpora a greater average precision compared to the lemma representation.

An interesting example to be worth noticing concern the Manhattan distance. The AP when using the token representation tends to decrease faster than the AP for the lemma representation as n increase.

The Euclidean distance seems to be the least appropriate distance measure for this task. The Clark distance have a really poor average precision when the MF tokens vector is too small, but after reaching 750 it produces one of the best payoff of the experiment.

Using the best n with these text representations, the best distance metric can increase the average precision by 13 to 17 %. Overall, the Cosine distance seem to be one of the most appropriate choice when dealing with these corpora and text representations.

The retained distance metrics and n-MF for the tokens text representations are:

- 750-MF tokens with Cosine distance.
- 750-MF tokens with Clark
- 750-MF tokens with Manhattan
- 750-MF tokens with Tanimoto

The $n = 750$ is used for every distance metric, since the results between 500 and 1000 are similar for most distance metrics.

Even though, the lemma representation sometimes give betters results than the token representation. Only the token representation is kept since every dataset come with a token representation and no significant improvements are observed for these datasets.

4.2.3 Text Size

For this study, an experiment on the St-Jean corpus is accomplished to show the importance of having large documents. To test this, the number of token is artificially modified by considering only the t first tokens of each text. Here t range between 9000 and 250 with steps of 250 tokens. From these texts, the 750-MF tokens text representation is used in conjunction with the Z-Score normalized Cosine distance to compute the rank lists.

Figure 3 shows the rank lists evaluated using the average precision, RPrec and HPrec over the number of tokens. The full evaluation is showed in Annex (Table H).

Every metric decreases over the text size. This indicates that it becomes harder to determinate documents pairs with the same author as the text size decrease.

PAN16 corpus is a difficult corpus due to its small size, thus extracting reliable features for each text to estimate each style is also a difficult task. After multiple tests, the PAN @ CLEF 2016 corpus is not used further in this study due to its difficulty in finding reliables stylometric clues. For this study, having standard and easier corpus is required in order to show the proposed methods.

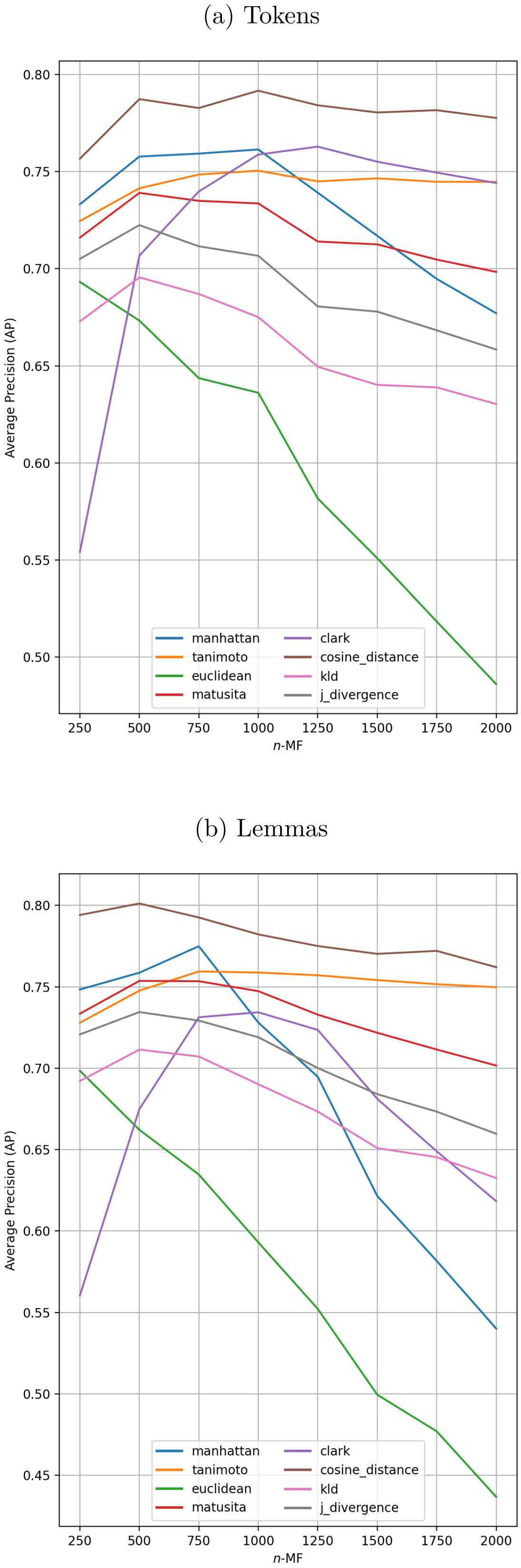
4.2.4 Most Frequent Items Restriction

As the state of the art suggested, the MF method proposed to limit the items used to only the most frequents in the vocabulary of a corpus. This experiment aim to show the importance of this limitation.

Three approach are compared:

1. *every token*: Use the whole corpus vocabulary to create the feature vector.
2. *without hapax legomena*: Every token appearing more than once are used to create these vectors.
3. 750-*MF*: 750 most frequent tokens are used to create the feature vector.

Figure 2: Tokens and Lemmas representation over number of MF tokens using different distances metrics

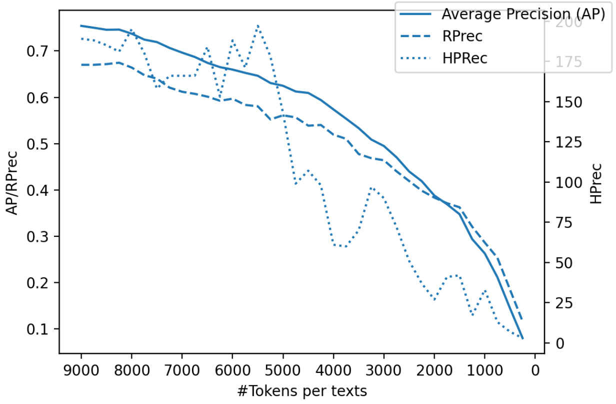


(a) Tokens

(b) Lemmas

The eight proposed distance measures are evaluated for the three approach using the average pre-

Figure 3: St-Jean ranks list evaluation on AP, RPrec and HPrec over the text size. Rank list computed using 750-MF tokens and the Z-Score normalized Cosine distance



cision metric on the three literature corpora. To understand more easily the results for the three approach, the *every token* approach is used as a baseline, from where the average precision gain is computed.

Table 2a shows the average precision baseline using the *every token* approach. Table 2c shows the average precision for the 750-*MF* approach. Table 2b and Table 2d show the gain over the baseline for the *without hapax legomena* approach and 750-*MF* approach respectively.

When looking at the baseline: the Manhattan, Euclidean and Clark distance measure have a poor average precision, all are below 0.3 in average.

The other distance metrics have medium to good results. It ranges from 0.58 for the KLD to 0.79 with the Cosine distance.

The Cosine distance with *every token* reach a 0.91 average precision for the Oxquarry corpus.

When removing hapax legomena, for most distance metrics the gain in average precision is limited, except for the Clark distance where the mean average precision rise from 0.3 to 0.5.

After limiting to the 750-MF, the Manhattan, Euclidean and Clark distance, the ones that had poor results for the baseline, all increase their average precision by around 48% which correspond to an average precision of 0.71, 0.63 and 0.78 respectively, which makes them in the same range as the other metrics in the baseline.

Across every metric and corpus, removing hapax legomena increase the average precision in average by 0.04, and limiting to the 750-MF by 0.21. This clearly indicate the importance of limiting the size of the vector for most distance metrics.

From this experiment, each distance metric can be placed in one of the two following categories: the ones sensible to large and noisy vectors, the ones not sensible. In the first category are: Manhattan, Clark and Euclidean, in the second: Cosine, Matusita and Tanimoto.

As for KLD and JD, they are neither in the first nor the second, since both of them had a small gain in regard to the Oxquarry corpus but larger gain for the St-Jean corpus. This, not so clear behavior, must be further investigated.

4.2.5 Frequent Errors

For this experiment, the goal is to try to understand the errors in the system, in this case the false links (document pairs with different authors) highly ranked on different rank lists. The rank list quality is high based on the text representation and the distance function, but mostly on the first. The previous statement can be deduced by the following reasoning: For example when using the n-MF method, if two documents feature vector are closely related (nearly identical), no matter the distance function they should have a low distance, since every distance function should give a distance of 0 when computing the distance between two identical vectors.

We believe some errors in the system are due to documents having close feature vectors values even though they are not from the same authors. To motivate this statement, the following experiment is realized.

The four token-based retained rank list from the St-Jean corpus are used, see Section 4.2.2. These four rank lists share the same text representation, in this case the relative frequency of the 750-MF tokens as feature vector, only the distance function used to create the rank lists are different. For the sake of this experiment, we define *frequent errors* as: *Links appearing in the top 20 false links of at least 3 out of the 4 rank lists.*

The goal is to compare the feature vector of two documents for specific links in the rank list. In our case, the links chosen are: Frequent errors, linked ranked first (most similar vectors, according to the distance function), link ranked last (the least similar vectors) and ranked HPrec-th in a rank list.

Table 2: Rank lists average precision depending on the number of token used

(a) Baseline: *every token*

Distance	Oxquarry	Brunet	St-Jean	Mean
Manhattan	0.27	0.21	0.17	0.22
Tanimoto	0.63	0.65	0.70	0.66
Euclidean	0.19	0.19	0.08	0.15
Matusita	0.61	0.63	0.62	0.62
Clark	0.40	0.28	0.21	0.30
Cosine	0.91	0.73	0.73	0.79
KLD	0.58	0.59	0.56	0.58
JD	0.59	0.63	0.59	0.60
Mean	0.52	0.49	0.46	0.49

(b) Gain *without hapax legomena* over baseline

Distance	Oxquarry	Brunet	St-Jean	Mean
Manhattan	+0.10	+0.05	+0.02	+0.06
Tanimoto	-0.00	+0.01	+0.00	+0.01
Euclidean	+0.11	+0.01	+0.00	+0.04
Matusita	-0.01	+0.01	-0.01	-0.00
Clark	+0.19	+0.16	+0.25	+0.20
Cosine	0.02	-0.02	-0.02	-0.01
KLD	-0.02	+0.02	-0.01	-0.00
JD	-0.01	+0.01	-0.02	-0.01
Mean	+0.05	+0.03	+0.03	+0.04

(c) 750-MF

Distance	Oxquarry	Brunet	St-Jean	Mean
Manhattan	0.67	0.68	0.76	0.71
Tanimoto	0.63	0.68	0.75	0.69
Euclidean	0.62	0.64	0.65	0.63
Matusita	0.63	0.68	0.73	0.68
Clark	0.89	0.72	0.74	0.78
Cosine	0.89	0.71	0.79	0.80
KLD	0.60	0.66	0.68	0.65
JD	0.61	0.67	0.71	0.66
Mean	0.69	0.68	0.73	0.70

(d) Gain 750-*MF* over baseline

Distance	Oxquarry	Brunet	St-Jean	Mean
Manhattan	+0.40	+0.47	+0.59	+0.49
Tanimoto	+0.00	+0.03	+0.05	+0.03
Euclidean	+0.43	+0.45	+0.57	+0.48
Matusita	+0.02	+0.05	+0.12	+0.06
Clark	+0.49	+0.44	+0.53	+0.48
Cosine	-0.02	-0.02	+0.06	+0.01
KLD	+0.02	+0.07	+0.12	+0.07
JD	+0.02	+0.04	+0.12	+0.06
Mean	+0.17	+0.19	+0.27	+0.21

To perform a comparison, the feature vector is visualized using a bar plot. The visualizations represent the frequency of the 750-MF tokens on the two feature vectors. To be able to understand more easily the visualization, the values have been sorted by the mean relative frequencies and use a logarithmic scale.

When a large proportion of the vectors overlap, it indicates a high similarity between the MF vectors. Both document style are close when their feature vector are closely related. Visually when most of the surface overlap, the distance function will give a low value, and with a correct distance measure this link should be ranked high in the rank list.

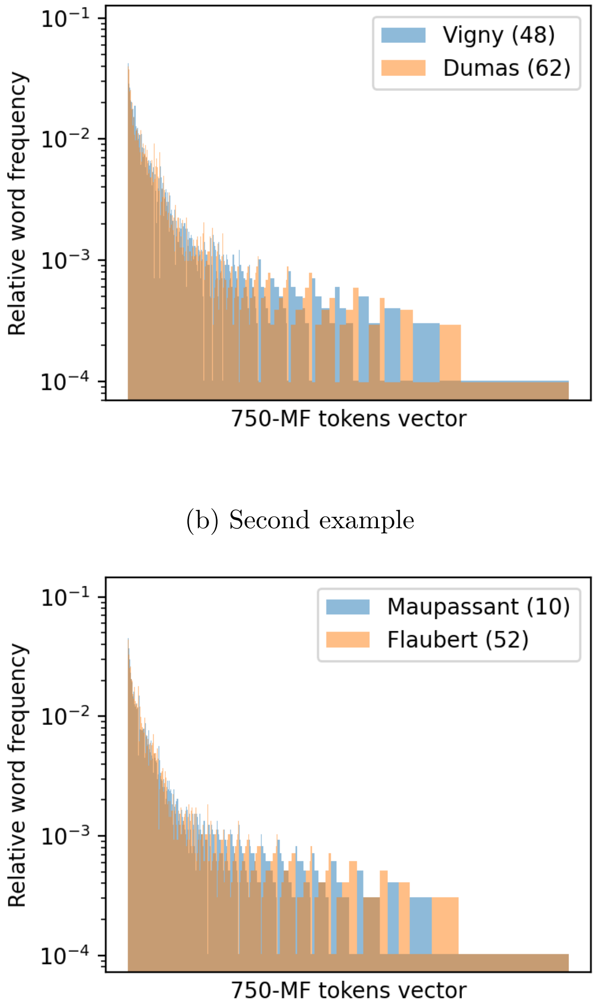
Figure 4 shows the feature vector of two frequent errors document pairs (*Cinq-Mars*$_{48}$ from Vigny / *Les trois mousquetaires*$_{62}$ from Dumas and *Bel-Ami*$_{10}$ from Maupassant / *Madame Bovary*$_{52}$ from Flaubert), these link appear in the top 20 false links of 3 out 4 rank lists. The frequent errors vector pairs presented in Figure 4 can be visually compared to actual true links and actual false links, to have a better understanding of this problematic. For example, the most similar true link (ranked 1 using Manhattan distance) in Figure 5a (*La Petite Fadette*$_{30}$ and *La Petite Fadette*$_{116}$ from Sand) or the HPrec-th (last continuous correct pair from the top of the list) in Figure 5b (*Les Diaboliques*$_{184}$ and *Les Diaboliques*$_{192}$ from Barbey), both of these links show a large proportion of overlapping surface, like for the frequent errors vectors.

A counter example would be the least similar false link (ranked last using Manhattan distance) which represent a dissimilar document pair. Figure 5c showcase this link (*Delphine*$_{183}$ from Stael / *La Double Maîtresse*$_{194}$ from Regnier). As expected, most of this figure surface is non-overlapping.

Since Figures 4 are closer to Figure 5a and Figure 5b than Figure 5c. We can assume that most distance function will not grasp a real difference and thus having them ranked high in the rank list.

When two vectors are relatively close together, determining that two texts are from a different author can not clearly be established using only

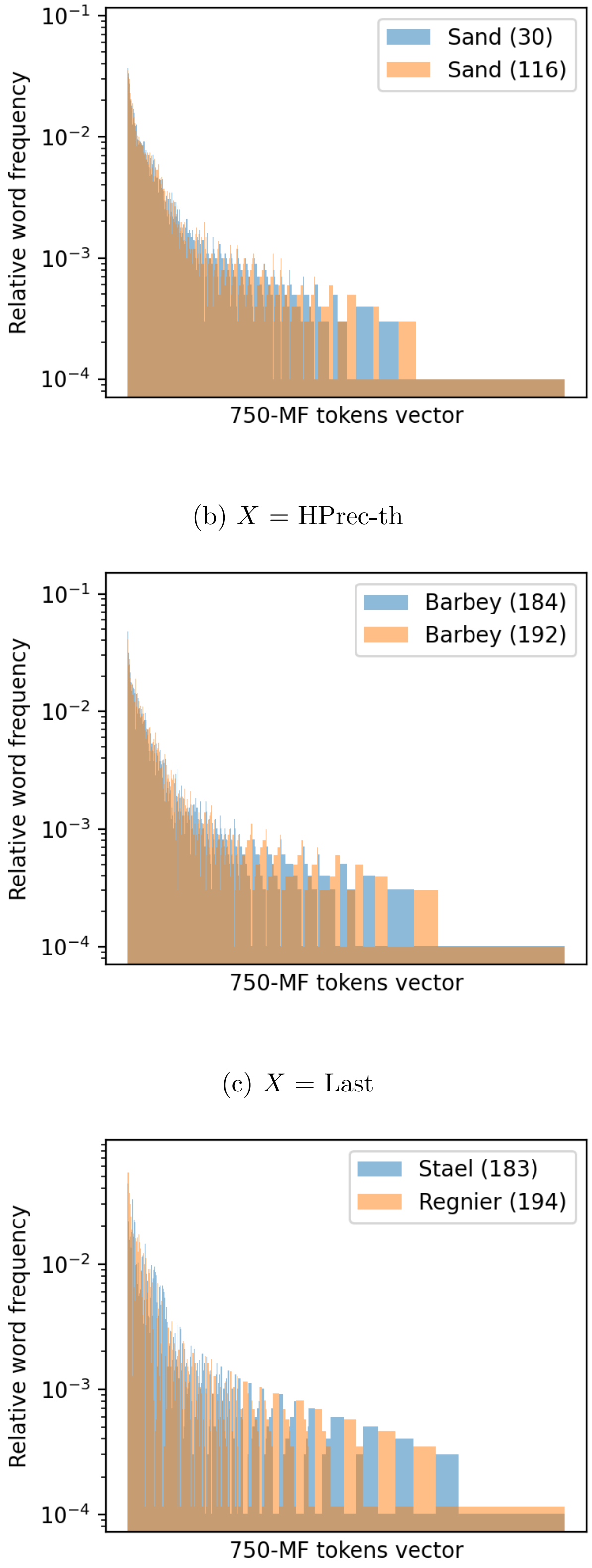
Figure 4: Example of 750-MF relative frequency vectors for recurrent false link in the top 20 false links



(a) First example

(b) Second example

Figure 5: 750-MF relative frequency for the two documents ranked X in the rank list using the token representation on St-Jean



(a) X = First

(b) X = HPrec-th

(c) X = Last

one type of representation, no matter the distance metric applied. Thus, this experiment motivate the need of having multiple text representation to obtain more robust solutions to be able to discriminate between true and false links for these types of errors.

4.3 MF Substrings

Instead of considering tokens or lemmas for the MF method, another strategy is to create short sequences of letters called substrings.

4.3.1 MF Letters n-grams

Letters n-grams is one possible substring method using Definition 30. The whole document is synthesized into a long string by joining every token with an *underscore*: "_". The underscore is used to remove visual ambiguity due to the nature of the space character. The n-grams representation aim to create items by considering part of words.

Definition 30 - Letters n-grams
A letter n-grams is a tokenization which is constructed by creating a token of size n for each substrings starting from the position 0 to $|\text{text}|-n-1$.
Example:
Using 3-grams the string: *"fox_is_running"*
is converted to: (*"fox", "ox_", "x_i", "_is", "is_", "s_r", "_ru", "run", "unn", "nni", "nin", "ing"*)

This technique can be further explored by combining, for example, two or more n-grams sizes together. In this study, the following notation is used: (a, b)-grams. This corresponds to both a-grams and b-grams in the same text representation.

n-grams can be really effective, even though no clear meaning can be exploited at first sight. A simple explanation on why n-grams can be effective is by using n-grams of size 3 to 5 in most Indo-European languages. With these sizes, it can capture the style of an author on part-of-speach or other stylistic aspects [2].

This can correspond for example to:

- In English, tense of the verbs: *"ing_", "ed_"*)
- In English, small common words: *"_the_", "_and_", "_that"*
- In French, adverbs : *"ment_"*
- In French, small common words : *"_le", "la_", "des_", "est_"*
- (...)

Table I and Table J in Annex show respectively the 40-MF 4-grams and the 40-MF 5-grams for the Oxquarry corpus.

When considering letters n-grams, the number of character in the n-grams text representation approximately increases by a factor of n compared to the token representation. To avoid having too large text representation, a possible idea is to remove the hapax legomena (tokens appearing only once in the corpus) before applying the n-grams algorithm. This method can also be generalized to remove words appearing less than x times. This pruning approach was explored in Kocher and Savoy [16] but was not used in this study.

4.3.2 MF In-word n-grams

An alternative version of the n-grams algorithm used here is defined in Definition 31.

Definition 31 - In-words n-grams
In-word n-grams are created by applying the letters n-grams algorithm on each token. When a token is smaller than n, the whole token is considered.
Example:
Using words 3-grams on the following tokens: ["fox", "is", "running"]
is converted to: (*"fox", "is", "run", "unn", "nni", "nin", "ing"*)

In this study, to differentiate it from the classical n-grams algorithm. The latter algorithm, is called *In-word n-grams* (Definition 31) and the other one is called *letters n-grams* (Definition 30).

This algorithm considers each token as a string to apply the letters n-grams algorithm to. In-word n-grams is a hybrid version between the letters n-grams and the word token text representations. This method yields a subset of the letters n-grams algorithm for the tokens larger than n.

4.3.3 MF n-First / n-Last

Two other similar approach to the n-grams are to consider only the n first letters of each token or the n last letters of each token, denoted respectively n-First and n-Last. Definition 32 and Definition 33 explain and propose an example. These methods are subsets of the in-word n-grams. These methods have the same number of items as the token text representation.

Definition 32 - n-First
The n-First algorithm create for every token a substring which is created from the first character to the n-th. If the token is smaller than n, the whole token is used.
Example:
Using 3-First on the following tokens: ["fox", "is", "running"]
is converted to: (*"fox", "is", "run"*)

The n-First correspond generally in Indo-European languages to the meaning of a word (i.e.: a part of the stem) This method can be related to the lemma approach.

Definition 33 - n-Last
The n-Last algorithm create for every token a substring which is created from the last character position minus n to the last. If the token is smaller than n, the whole token is used.
Example:
Using 3-Last on the following tokens: [”fox”, ”is”, ”running”]
is converted to: (*”fox”, ”is”, ”ing”*)

The n-Last correspond generally in Indo-European languages to, for example, tense for verbs or more generally to suffixes. The method can also be related to the part-of-speech method.

4.3.4 MF Letters n-grams Evaluation

To evaluate the n-grams with the MF approach, another methodology is used. Since the n-grams create an additional parameter, the field of possible parameter increase and induce more computations.

To reduce the field of the possible parameters, the experiment is split in two parts: The first part aim to find the optimal n-grams size, and the size of n-MF n-grams vector. The second is to compare the distance metrics with the previously found parameters.

One of the main drawback with this methodology is that some distance metrics can have better performance on highly dimensional vectors (large MF vector) than others. Or some n-grams size have better results depending on the distance metric. The current methodology does not take this into account.

First part

For the first part, the Z-Score normalized Cosine distance measure is used to compare the vectors and create the rank list. The parameters used are:

- A varying n-MF vector size ranging between 500 and 15000 with a step of 500.
- n-ngrams with n = 3, 4, 5, $(2, 3)$, $(3, 4)$, $(4, 5)$.

The number of different n-grams is assumed to be larger than the vocabulary of the texts. Thus, having a feature vector larger than the one used for the token and lemma text representation is advised.

Figure 6 shows the average precision for the experiment first part on the Oxquarry corpus (6a), Brunet (6b) and St-Jean (6c).

The trends seem to indicate that the smaller the n-grams are, the smaller the n-MF vector need to be to converge to a maximal effectiveness. This can be explained by the fact that the vocabulary of the large n-grams is larger than the ones with small n-grams. Thus it requires more MF items to discriminate the authors styles.

The $\sim 3000 - 4000$-MF 3-grams give good results on Oxquarry and Brunet.

On the Brunet corpus, after reaching the maximal average precision at $\sim$ 4000-MF with 3-grams, a larger MF decrease the precision until it converges at $\sim$ 6500-MF. For Oxquarry with 3-grams, the average precision increase until $\sim$ 4000-MF but stay still after this. On St-Jean, the 3-grams and $(2, 3)$-grams do not reach a high average precision with small n-MF like for the Brunet and Oxquarry.

4-grams also can be efficient but require a larger MF vector, around $\sim$ 4000-MF give good result for both Brunet and St-Jean but on Oxquarry it requires around $\sim$ 8000 to have similar results as the 3000 3-grams.

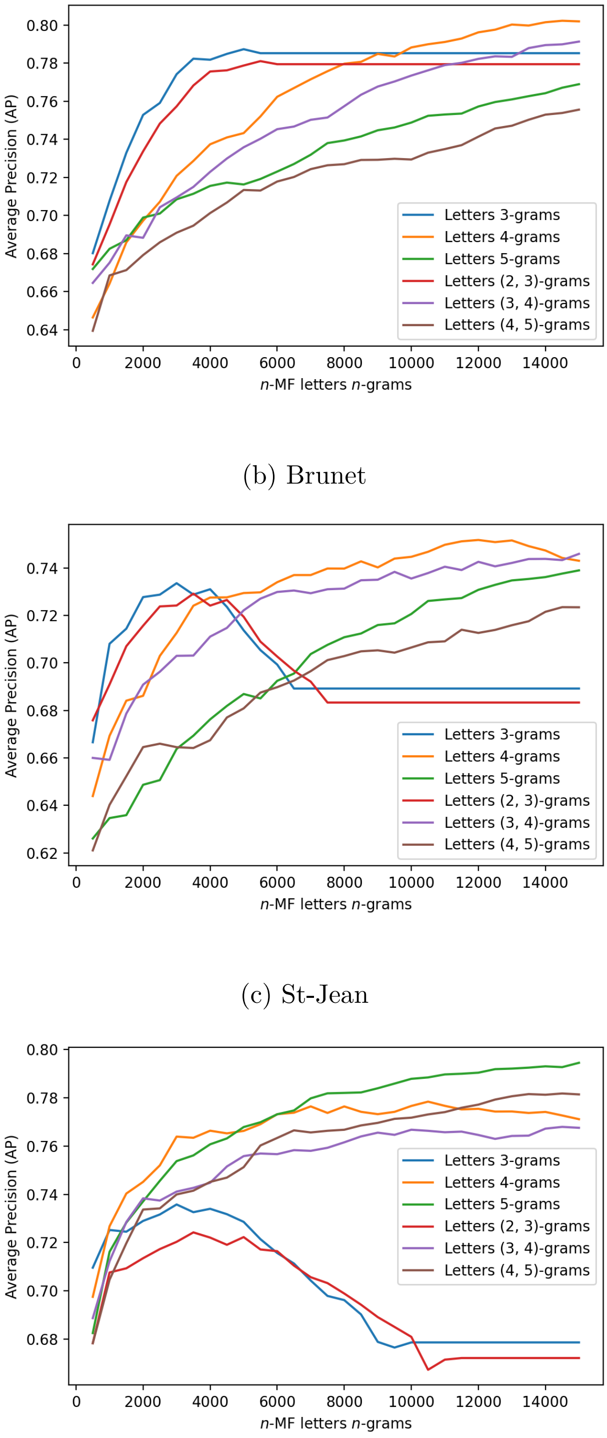
For the first part, the retained parameters for the n-grams text representation are the 3-grams with 3000-MF and 4-grams with 8000-MF.

Second part

As stated earlier, in the second part the objective is to compare the distance metrics with the 3-grams/3000-MF and 4-grams/8000-MF. The distance metrics used are the ones presented in Section 3.5. The results for this experiment are in Table 3. Since the combination of n-MF and n-grams size was optimized for the Cosine distance, the results may be biased.

For the 3-grams, the following results can be observed: The Cosine distance give the best results. Though, two distances metrics can compete with the Cosine distance on some corpora with the 3-grams approach, namely the Manhattan distance and the Clark distance. A worth noticing results, is the Euclidean distance, which have polarized results: on Oxquarry the average precision is 0.72 and on St-Jean only 0.47. Using the right distance measure can have a relative increase up to 57%, when comparing the worse distance metric

Figure 6: Letters n-grams representation over number of MF with different n



(a) Oxquarry

(b) Brunet

(c) St-Jean

to the best for each dataset. The 57% increase is obtained by using the Cosine distance (0.74) instead of Euclidean distance (0.47) with the St-Jean dataset.

Table 3: Average precision for n-grams with every metric, on the three corpora (best result for the dataset in bold).

(a) 3-grams with 3000-MF

Distance metric	Oxquarry	Brunet	St-Jean
Manhattan	**0.77**	0.66	0.62
Tanimoto	0.64	0.66	0.62
Euclidean	0.72	0.63	0.47
Matusita	0.60	0.66	0.58
Clark	0.64	**0.73**	0.63
Cosine	**0.77**	**0.73**	**0.74**
KLD	0.57	0.65	0.54
JD	0.58	0.66	0.56

(b) 4-grams with 8000-MF

Distance metric	Oxquarry	Brunet	St-Jean
Manhattan	0.76	0.69	0.73
Tanimoto	0.68	0.68	0.68
Euclidean	0.73	0.65	0.55
Matusita	0.63	0.68	0.65
Clark	0.67	0.72	0.75
Cosine	**0.78**	**0.74**	**0.78**
KLD	0.61	0.66	0.60
JD	0.61	0.67	0.63

For the 4-grams approach, as for the 3-grams the Cosine is the best measure with this configuration. The Manhattan distance and the Clark distance also are the second choice. The Euclidean distance have the same behavior as for 3-grams on the dataset. For this configuration of 4-grams with 8000-MF, using the right distance measure is less impactful than the 3-grams approach. A relative increase in performance up to 42% can be observed when changing from the worse distance metric to the best for each corpus. The 42% increase is also obtained by using the Cosine distance (0.78) instead of Euclidean distance (0.55) with the St-Jean dataset.

4.3.5 MF In-word n-grams / n-First / n-Last Evaluation

In this experiment, the goal is to compare the three following substring text representations: In-word n-grams, n-First and n-Last.

To reduce the field of parameters, only the Z-Score normalized Cosine distance used and is evaluated using the average precision. For this experiment, the number of n-MF variate between 200 and 4000 with a step of 100. The n for the In-word n-grams, n-First and n-Last variate between 3 and 5.

Figure 7 shows the average precision for this experiment on the three corpora (Oxquarry, Brunet, St-Jean).

In the three dataset, the 3-Last and 3-First gives a lower average precision and quickly converge to an equilibrium value at around 1500-MF. For the three methods, when $n = 5$, it tends to produce better results than $n = 3$ or $n = 4$ as the n-MF increase. In the Oxquarry corpus, in-word 4-grams and in-word 5-grams give an outstanding $\sim 95.0\%$ in average precision. But this method is not as effective for the Brunet dataset. For the Brunet dataset, the 5-First have good results compared to other text representations with small MF vector size. Lastly, for St-Jean the 5-First with a small MF vector size give the best results.

No clear best configuration can be extracted with these results. Results are mixed across corpora, thus for this experiment, no text representation / distance measure are retained since we aim to find high quality and consistent methods.

4.4 MF POS Sequences

Another possible stylistic aspect that can be detected is to consider the sentence constructions. A part-of-speech (POS) tagging algorithm is required to apply the technique proposed on any text. In this study, we will not cover this aspect and assume that the POS are available and correctly represent the texts.

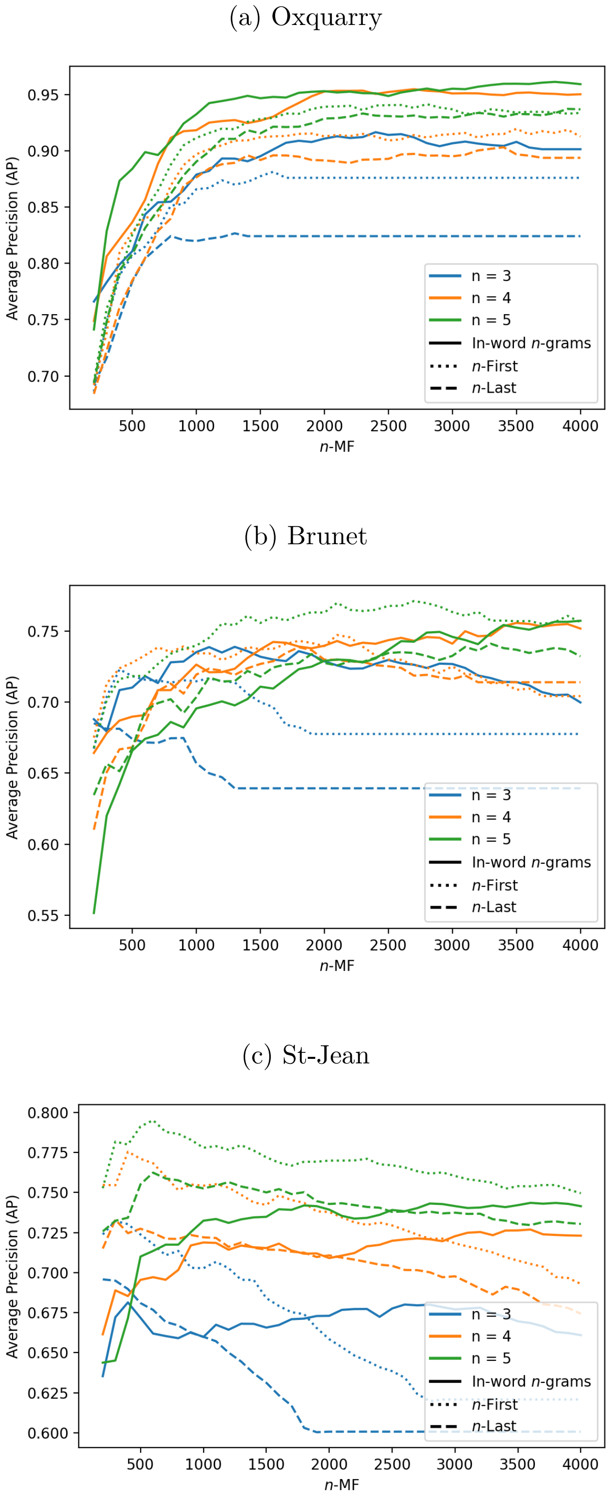
4.4.1 Method

The idea here is to create a text representation based on POS sequences. Used in conjunction with the MF method, this can produce feature vectors which can be used to compare documents.

These short sequences are created using the same principle as for the n-grams. From a list of POS, each POS is considered as a letter when applying the n-grams algorithm. This type of n-grams is also known as w-shingling.

Definition 34 - POS Short Sequences
Each word is tagged with a corresponding POS. The document is represented as a long ordered list of POS. From the list of POS, a moving window of size n, creates the POS sequences. For the sake of simplicity, in this study, a POS sequence of size n is written n-POS.

Figure 7: Average precision over the n-MF In-word n-grams, n-First and n-Last using Z-Score normalized Cosine distance



(a) Oxquarry

(b) Brunet

(c) St-Jean

The following example showcases a sentence in the 3-POS (POS sequences of size 3) text representation.

Sentence	*"The", "cat", "eat", "a", "fish"*
POS	*"Article", "Noun", "Verb", "Article", "Noun"*
3-POS sequences	*"Article Nounp Verb", "Noun Verb Article", "Verb Article Noun"*

In practice the POS are more detailed, for example instead of just considering *eat* as a verb, a more detailed POS can be the verb and its tense *Verb-SimplePresent*, the same goes for the other type of POS.

The MF method can be applied to the n-POS text representation to turn each document into a feature vector. Then they can be compared using distance metrics to find distance between documents.

4.4.2 Evaluation

For this experiment, n-POS are used to detect the style of the author.

1-POS are discarded from the experiment (equivalent of the POS frequency) since previous studies showed that 1-POS tends to produce worse results than with larger n [16].

To lower the computational cost, this evaluation is split in two parts, the first aim to select the most convincing size of n-POS and the best fitting n-MF. The second is to compare the different distance metrics. Keep in mind, as for the previous experiment in two parts, this experimentation methodology ignore the strength and weakness of distance measure with regard to the dimensionality of the vectors.

First part

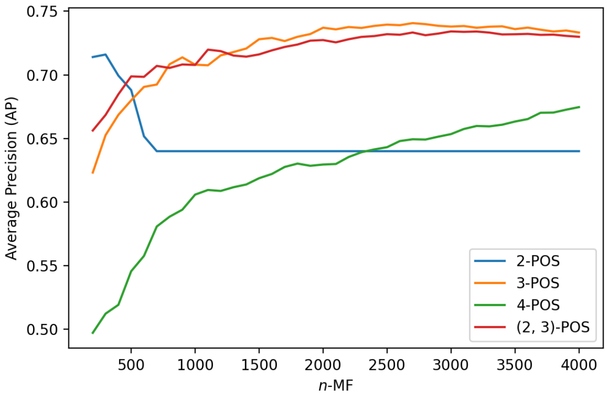
In the first part, 2-POS, 3-POS, 4-POS and the combination of the 2-POS and 3-POS denoted: $(2,3)$-POS are used. The distance metric used for this part is the smoothed Z-Score normalized Cosine distance. For this representation no clear n-MF vector size is advised, the size used is between 200 and 2000 with a step of 100.

Figure 8 shows the average precision on the rank list produced by using n-POS over the n-MF.

The two following information can be intuitively observed on this plot:

- The $(2,3)$-POS have similar results to the 3-POS method. Due to its larger text representation, the $(2,3)$-POS can already be discarded.
- A larger / more complex n-POS require a larger n-MF to achieve its maximal effectiveness. In the St-Jean corpus, a total of 26 different POS are used to describe every word in the corpus. Which correspond to $26^2 = 676$ possible unique 2-POS, to $26^3 = 17,576$ 3-POS and $26^4 = 456,976$ 4-POS. Thus the 2-POS converges on this plot but not 3-POS and 4-POS.
- Like other methods, if the n-MF is too large, an overfitting to less important items is possible, thus reducing the average precision. In Figure 8 the 2-POS clearly have a drop in average precision after $\sim$ 250-MF.

Figure 8: Average precision over the n-MF for the rank list generated using the Z-Score normalized Cosine distance on St-Jean's n-POS text representation.



Using the smoothed Z-Score Cosine distance, the most appropriate configuration for the n-POS text representation on St-Jean is 2-POS with 250-MF and 3-POS with 1000-MF.

Second part

In this second part, the goal is to find the most appropriate distance metric for the n-POS text representation, with the configuration retained in the first part. After running with these parameters on every proposed distance metrics, Table 4 shows a resume of the experiment.

When considering n-POS as text representation on the St-Jean corpus, overall the best distance metrics are Manhattan, Matusita, JD, Cosine distance. The Clark distance, give overall the worse results, even though in other text representation,

Table 4: Average precision for every distance metrics with the n-POS representation on St-Jean

	n-MF/n-POS	
Distance metric	250-MF/2-POS	1000-MF/3-POS
Manhattan	0.70	**0.75**
Tanimoto	0.67	0.72
Euclidean	0.69	0.72
Matusita	0.70	0.73
Clark	0.50	0.60
Cosine	**0.73**	0.71
KLD	0.68	0.70
JD	0.69	0.73

this metric was giving the best results.

Optimizing the parameters with this text representation can increase the average precision by up to 46%. This increase is obtained by using the Cosine distance instead of the Clark distance for the 250-MF/2-POS.

Two configurations are retained using the n-POS text representation. The first is the 2-POS with the 250-MF and the Z-Score Cosine Distance. The second is 3-POS with the 1000-MF and the Z-Score Manhattan distance.

4.5 Compression-based Distances

This section covers another method to compute distances between documents based on file compression.

4.5.1 Method

The main idea is to first compress two documents A, B then compress the concatenation of A and B, denoted AB.

Using the sizes after compression of A, B, AB and a compression distance measure, it is possible to compute a distance between A and B.

For the compression, a lossless compression algorithm is used. The Lempel-Ziv family (GZIP), the block sorting family (BZip2) and the statistical family (PPM) have been experimented in Oliveira and al. [6] and show good results.

This technique is based on the fact that lossless compression algorithms tries to lower the Shannon entropy of a document. When compressing a document with a large Shannon entropy, the compressed document should have a larger size after compression than a document with a small Shannon entropy. When concatenating two documents that share many terms, the entropy of the concatenated document should be lower than if the two documents present distinct vocabulary.

This approach has the benefit to produce rank lists in a nearly parameterless manner, only a distance metric and compression algorithm are needed. The main drawback with this technique is the fact that the results/decisions can not be entirely explained.

In this study the lossless compression algorithm used are: GZip, BZip2, LZMA. The implementations for these compression algorithms are the ones in the Python standard library, the programming language used for this study [30].

Each compression algorithm can be tweaked with different parameters, the default settings are used except for the compression level (trade-off compression time and compression size). We set the compression level to the maximal setting. This allows to ensure that the produced file will have the lowest possible Shannon entropy reachable with this algorithm. Thus providing the best possible approximation of distance when used in conjunction with the compression distances.

Definitions 35, 36 and 37 are compression distance measure found in the literature [6] [24].

Definition 35 - Conditional complexity of compression [6] [24]
The conditional complexity of compression of two documents A and B is computed as follows:

$$CCC(A, B) = C(AB) - C(A)$$

C(AB) is the size after compression of the concatenation of A and B
C(A) the size after compression of A.
This metrics is not easy to use since the order of magnitude is not bounded and can depend a lot on the text sizes. The next ones try to mitigate this problem.

Definition 36 - Normalized compression distance [6] [24]
The normalized compression distance of two documents A and B is computed as follows:

$$NCD(A, B) = \frac{C(AB) - \min(C(A), C(B))}{\max(C(A), C(B))}$$

C(AB) is the size after compression of the concatenation of A and B
C(A) the size after compression of A
C(B) the size after compression of B.
This metric gives a value in the range $[0, 1 + \epsilon]$, with ϵ being a small positive value created by the imperfection of compression algorithms.

Definition 37 - Compression-based cosine [6] [24]
The compression-based cosine of two documents A and B is computed as follows:

$$CBC(A, B) = 1 - \frac{C(A) + C(B) - C(AB)}{\sqrt{C(A) \cdot C(B)}}$$

C(AB) is the size after compression of the concatenation of A and B
C(A) the size after compression of A
C(B) the size after compression of B.
This metric has the same bounds as the cosine distance (Definition 13).

4.5.2 Evaluation

An experiment was conducted to try to compare the three proposed compression algorithm (GZip, BZip2, LZMA) for the compression based distance ranking.

For each document, the size after compression is computed with each algorithm. We also compute the size after compression of the concatenation of each document pair.

Using these sizes and the NCD or CBC distance metrics (ref. Section 4.5), the rank list are computed and evaluated. The results in terms of efficiency of the resulting rank list are shown in Table 5.

The experiment is run on the three corpora three times to have a better approximation of the run time. The average time of the three runs are in Table 6. The CPU used for the experiment is an *Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz*.

LZMA gives the best results on every corpus tested (Mean AP: 0.76 with NCD and 0.78 with CBC). BZip2 have results close to LZMA (LZMA have an AP $\sim 2.5\%$ better). GZip give the worse results on every corpus. It obtains an AP $\sim 30\%$ worse than LZMA.

The Cosine-based compression distance (CBC) tend to give slightly better results over the normalized compression distance (NCD). The AP is $\sim 1\%$ better with the CBC.

In terms of time complexity, BZip2 is the fastest algorithm of the three proposed. GZip is slower than BZip2 by around $\sim 7\%$. LZMA is $\sim 5-6$ times slower than BZip2.

No significant time differences are recorded between the NCD and CBC distance measures. This is explained by the fact that, the greatest complexity reside in the compression algorithm.

Previous study show that the distance measure choice was more impactful than the choice of the compression algorithm in the regarding the quality of the results [6]. Our results tend to indicate the opposite, ref. Table 5. The distance measures does not have a large impact on the quality of the results. Though the compression algorithm does. This difference may be caused by the difference of compression algorithm used between the two experiments.

For this text representation, the retained configuration is the BZip2 algorithm with the CBC distance measure. Even though the LZMA algorithm give the best results, we decided to trade quality for time. BZip2 produces relatively good results in a shorter amount of time ($\sim 5-6$ times faster).

4.6 Individual Methods Summary

For the next experiments of this study, medium to high quality rank list are required. In this chapter, we selected methods that can produce qualitative rank lists using different text representations.

Figure 9 contains a schema summarizing the steps to obtain rank lists using the different approaches.

The nine retained methods are presented in Table 7. Four of them are using the n-MF tokens, two n-MF tokens n-grams, one compression techniques and two the n-POS with n-MF.

The complete evaluation of the rank list produced

Table 5: Compression methods evaluation with different compression algorithm and distance metrics

(a) Distance: NCD

AP/RPrec/HPrec	Oxquarry	Brunet	St-Jean	Mean
Bzip2	0.77/0.68/69	0.76/0.70/25	0.70/0.63/214	0.74/0.67/102
GZip	0.62/0.56/41	0.61/0.53/24	0.45/0.44/054	0.56/0.51/040
LZMA	0.81/0.70/82	0.78/0.73/27	0.71/0.63/241	0.76/0.68/117

(b) Distance: CBC

AP/RPrec/HPrec	Oxquarry	Brunet	St-Jean	Mean
Bzip2	0.79/0.69/74	0.76/0.70/25	0.70/0.62/219	0.75/0.67/106
GZip	0.64/0.56/43	0.60/0.52/23	0.42/0.42/056	0.55/0.50/041
LZMA	0.84/0.73/85	0.79/0.73/31	0.71/0.62/214	0.78/0.69/110

Table 6: Compression methods time evaluation with different compression algorithm and distance metrics

(a) Distance: NCD

Time	Oxquarry	Brunet	St-Jean	Mean
Bzip2	12.7s	8.4s	198.9s	73.3s
GZip	15.0s	8.8s	211.3s	78.4s
LZMA	69.0s	46.6s	1046.3s	387.3s

(b) Distance: CBC

Time	Oxquarry	Brunet	St-Jean	Mean
Bzip2	12.7s	8.4s	198.4s	73.2s
GZip	14.9s	8.9s	214.5s	79.4s
LZMA	68.8s	46.8s	1052.0s	398.2s

on these text representation for the three literary corpora are presented in Annex (Table K and Table L).

4.6.1 Distance Matrix Visualization

From a rank list, it is possible to create a distance matrix (ref. Section 3.6.3). Distance matrices created from rank list can be visualized. To represent distance matrices, each element of the matrix is mapped to a pixel in an 2D image. The element value in the matrix is mapped to the pixel brightness. The low values are in light colors and large values in dark colors

A good distance matrix should have each same author documents pair with a low distance (light color in the image) and different authors documents pairs with a large distance (dark color in the image). The greater the contrast between the true links (same author documents) and the false links is (different authors documents), the better the distance matrix is.

To better understand more easily this matrix, authors are sorted alphabetically. If the distance matrix can represent correctly the author style, same authors documents should have a low distance (light color in the image). This should create light color squares in the diagonal. The square size is related to the number of document written by this author. The diagonal is the lightest color (white), since the distance between two same documents is always 0, with respect to the distance functions identity of indiscernible axiom.

The distance matrix for the best retained representation (the largerst AP) and the worse retained text representation (the lowest AP) is visually presented in Figure 10 for Oxquarry using a blue tint. Respectively, the Clark distance on the 750-MF which gives an average precision of 0.89, and the Tanimito distance on the 750-MF gives an 0.63 average precision. The diagonal is white in both images. Light colors squares in the diagonal can clearly be observed on both distance matrix visualizations. Some are slightly tainted, these documents pairs are harder to assert to be of a same authorship.

The Clark distance has overall a good distances matrix, except some document pairs from *Conrad*, *Hardy* and *Orczy* which have darker colors, these are ranked below some false links in the rank list.

For Tanimoto, firstly one can observe these

Figure 9: Rank lists methods schema

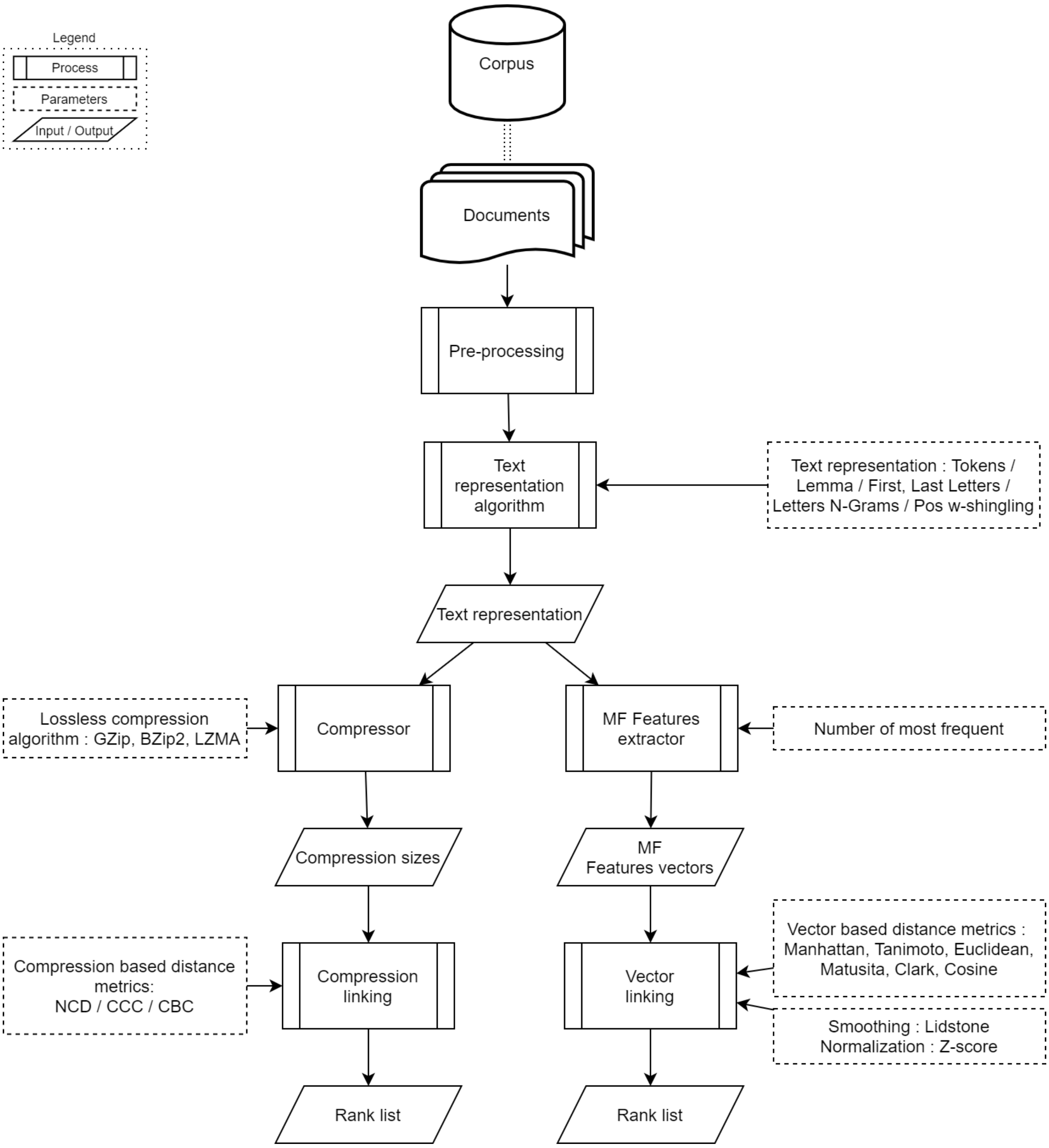

Legend
Process
Parameters
Input / Output
Corpus
Documents
Pre-processing
Text representation algorithm
Text representation : Tokens / Lemma / First, Last Letters / Letters N-Grams / Pos w-shingling
Text representation
Lossless compression algorithm : GZip, BZip2, LZMA
Compressor
MF Features extractor
Number of most frequent
Compression sizes
MF Features vectors
Vector based distance metrics : Manhattan, Tanimoto, Euclidean, Matusita, Clark, Cosine
Compression based distance metrics: NCD / CCC / CBC
Compression linking
Vector linking
Smoothing : Lidstone Normalization : Z-score
Rank list
Rank list

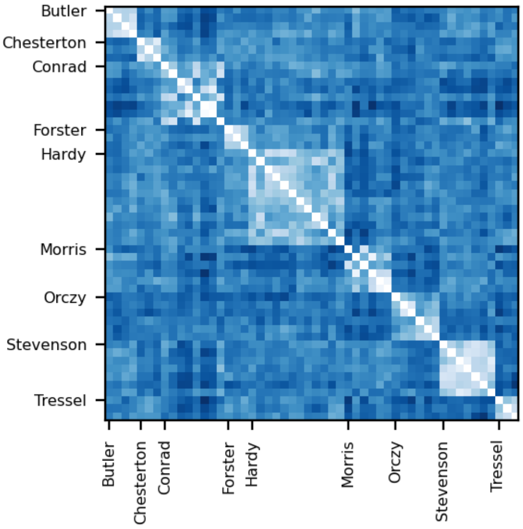
Table 7: Retained text representation and configuration

Id	Text representation	Distance measure	Z-Score	Lidstone λ
0	750-MF tokens	Cosine Distance	Yes	10^{-1}
1	750-MF tokens	Clark	No	10^{-1}
2	750-MF tokens	Manhattan	Yes	10^{-1}
3	750-MF tokens	Tanimoto	No	10^{-1}
4	3000-MF tokens 3-grams	Cosine distance	Yes	10^{-1}
5	8000-MF tokens 4-grams	Cosine distance	Yes	10^{-1}
6	BZip2 compression	CBC distance	No	10^{-1}
*7	250-MF 2-POS	Cosine distance	Yes	10^{-1}
*8	1000-MF 3-POS	Manhattan distance	Yes	10^{-1}

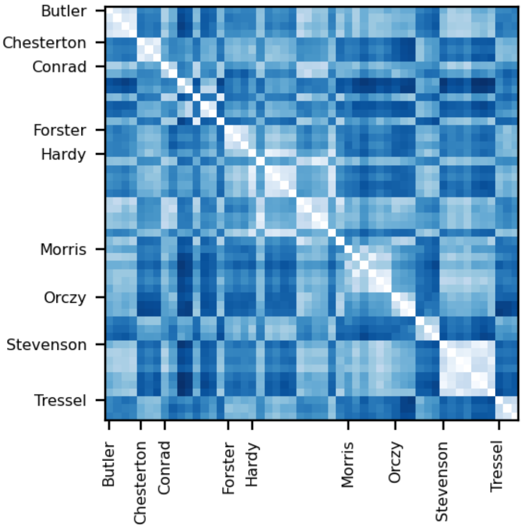
Note: Text representation and configuration with a star (*) are only used for the St-Jean corpus.

Figure 10: Distance matrix visualization using Oxquarry

(a) Best retained text representation for Oxquarry (750-MF tokens with Clark)



(b) Worse retained text representation for Oxquarry (750-MF tokens with Tanimoto)



stripes, which are due to the max function in its computation, which create a more cleaved decision in the score value.

4.6.2 Publication Date

When dealing with false links ranked high in the rank list, as the previous experiment showed, some of these excerpt pairs use similar words (ref. Section 4.2.5). These shared words might be related to the era the book was written in. The following experiment tries to investigate on this assumption. We try to verify this assumption by analyzing the difference in publication date for the most incorrect document pairs in the rank lists (false links ranked high in the rank list).

In the St-Jean corpus publication paper, the publication dates of each excerpt are available [14]. First, the publication date distribution of the corpus must be understood. Figure 11a show the distribution of the publication date in the St-Jean corpus. The corpus mainly focus on the two last thirds of the XIX century, only a few books are published between 1800 and 1830.

The date difference distribution for each pair of documents can be computed. Figure 11b shows the date difference distribution for the true and false links. The union of both of them represent every possible document pairs (the whole rank list), in this figure the bars are stacked to represent every link. Table 8 shows statistics on the distributions.

True links have a low mean date difference of 5.11 years with a standard deviation of 7.05 years. This low mean can be explained by the fact that most authors in the St-Jean corpus have excerpts from the same book. Also, the authors publish their books during their career, which is limited to their active years (life span minus early stages and old age for most authors). In fact, 281 of the 670 ($\sim$ 42%) true links are excerpts published the same year. 453 out of 670 ($\sim$ 68%) true links have a publication date difference below or equal to 5 years.

For St-Jean the largest date difference from the same author (correspond to the longest career in the corpus) is 31 years, which correspond to the two following books from Victor Hugo: *Notre Dame de Paris* (The Hunchback of Notre-Dame), in 1831 and *Les Misérables*, in 1862.

The average date difference for the false links is 28.24 years, with a standard deviation of 20.73 years. The overall average (both true and false links) is at 29.04 years, with a standard deviation of 20.58 years.

The previous statistics can be compared to the ones in Figure 11c. This figure shows the date difference density on the top-r false links (r is chosen at 670, it corresponds to the number of true links in St-Jean) in a rank list. The rank list used have an average precision of 85%. It is obtained by the Z-Score fusion of the retained text representations

Table 8: Date differences statistics

Links	Mean	Std
True links	5.11	7.06
False links	29.04	20.58
True and False links	28.24	20.73
top-r False links	21.29	16.00

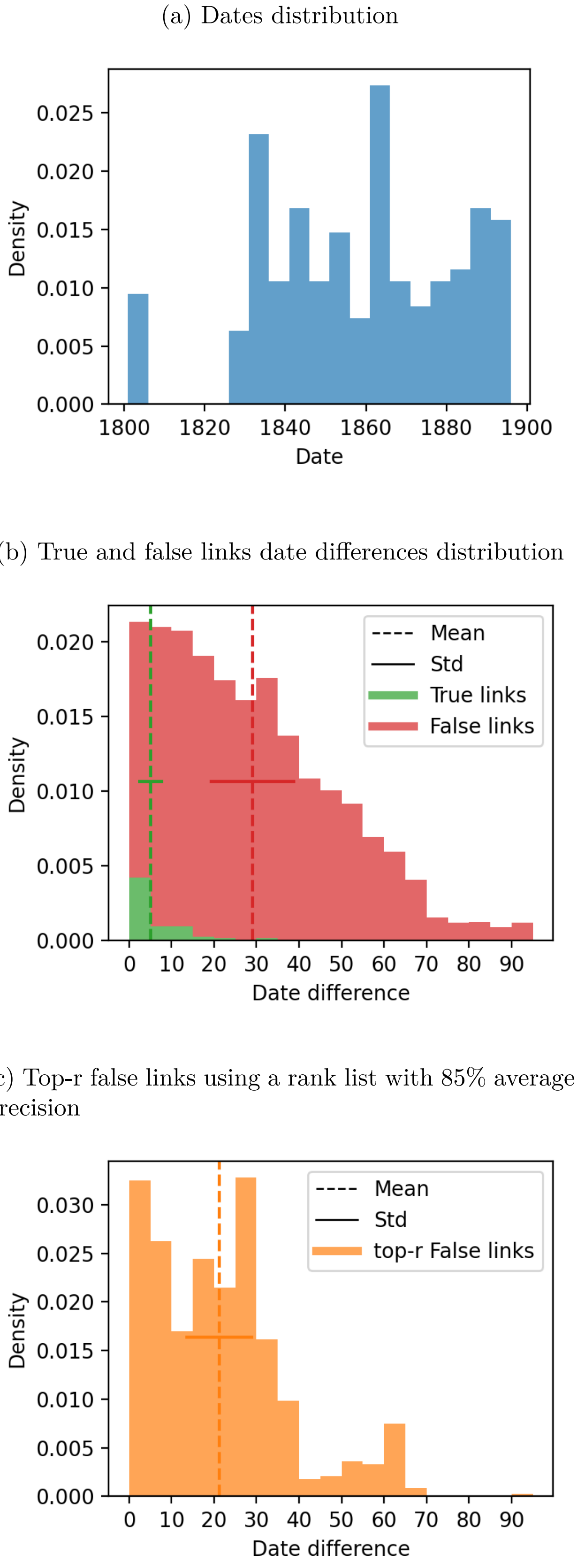
(Table L, ref. Chapter 6).

Two interesting information can be extracted here. Firstly, the mean is lower by 7.75 years (29.04 - 21.29) compared to every false links and have a narrow standard deviation distribution. Having a lower mean in this case indicate that more mistakes are made for document pairs with a lower date difference then for the ones with a large date difference, which can lead to the following conclusion: distinguishing between a false link and a true link for documents pair with low date difference is harder. Secondly, we can observe a drop after 35 years of date difference, which indicate that links in the interval [0 − 35] years are harder to discriminate between a true link and a false link than the ones outside this interval.

The 35 years interval can be related to the generation factor, the age of woman giving birth is around 25-34 in France [18], authors' birth country for this corpus. Each new generation tends to use its own vocabulary, and thus it can be harder to discriminate the author of text belonging to the same generation, if we assume that the authors write their books at around the same age. In the other hand, having different vocabulary can indicate a different time period and can be used to detect document forgery [24].

The small spike between 60 and 65 in the top-r false link is due to the matching of most excerpts from *Volupté* (excerpts 117, 135, 151, 165, 181, 189) written by Charles Sainte-Beuve in 1834 and *Les plaisirs et les jours* (excerpts 114, 132, 148) written by Marcel Proust in 1896. Out of the 18 possible false link for these excerpts, 12 are in the top-r false link. This indicates that the styles used in these two books are close. We can not discriminate the authors correctly even though the books were published with 62 years interval. This interesting observation could be further discussed with a historian specialized in literature.

Figure 11: Dates distribution and date differences distribution on St-Jean



(a) Dates distribution

(b) True and false links date differences distribution

(c) Top-r false links using a rank list with 85% average precision

Chapter 5

Authorship Clustering

In this chapter, authorship clustering methods will be presented and evaluated.

As explained in Section 3.7, authorship clustering aim to regroup documents into clusters. The best achievable clustering have every same author document in the same cluster and every different author document in a different cluster. The quality can be measured with the metrics presented in Section 3.7.2.

For the clustering part, multiple methods based on hierarchical clustering are proposed and evaluated:

- Silhouette-based clustering: A fully unsupervised clustering which aims to minimize the intra-cluster distances and maximize the nearest cluster distance.
- Distribution-based clustering: A supervised clustering technique which aims to model known corpus rank lists and compute a decision point.
- Regression-based clustering: A supervised clustering technique which takes clues from previous corpora analysis to lead the clustering.

The hierarchical clustering is the simplest connectivity model to achieve clustering from a distance matrix. As explained in Section 3.6.3, rank lists and distance matrices are related, thus after computing rank lists this clustering method seems to us to be the right choice. Additionally the hierarchical clustering model can use any distance measure which allow flexibility.

The main draw back of this method is the computation cost to compute the rank lists / distance matrices ($O(n^2)$). This problem is partially ignorable since this study focus on texts which are easily to handle, for example when compared to images.

5.1 Hierarchical Clustering

To find clusters of authors, a possible way is to use a hierarchical clustering algorithm on a rank list. The rank list indicate with a score if the two documents should belong to the same cluster. The hierarchical clustering regroup documents following this order.

The hardest task with this clustering scheme is to find the position in the rank list where true link become less frequent than false link. In the clustering context, the optimal position is unknown since the true labels are not available. We call this type of problem unsupervised. The position should minimize the true links under it and the false links above it.

In this study, finding this position will also be referred to as finding the *cut*. We define the true positive as true links above the cut, true negatives as false links under the cut. In addition, false positive are true links under the cut and false negatives are false links above the cut.

To find this cut, three approaches are explored: a Silhouette-based model, a distribution-based model and a regression-based model.

The Silhouette-based method can be used when only one corpus is available and no learning procedures can be applied. This method is presented and evaluated in Section 5.2.

The distribution-based model learn based on a corpus rank list with labels (training corpus) a score where the cut should be for any new corpus rank list (called testing corpus). Section 5.3 contains explanations for this method.

Finally, the regression based model, estimate for each links in the rank list, the probabilities of being true links. This model is trained on a corpus with labels (examples, training corpus). The model can find the optimal cut for a new corpus rank list (called testing corpus), by estimating the true link probabilities for each link. This approach is explained in Section 5.4.

5.1.1 Algorithm and Implementation

The scikit-learn Python module [34] provide a bottom-up implementation of the hierarchical clustering, which is called agglomerative clustering.

The agglomerative clustering follow this procedure:

1. The rank list used is converted into a 2D distance matrix, with each link representing an element of the matrix (ref. Section 3.6.3).
2. At the beginning, each document is considered as a single document cluster.
3. The link (element) with the lowest score in the matrix define the two documents to be merged.
4. The matrix is updated following a linkage criterion. Columns and rows of the clusters merged are updated by appling a linkage criterion.
5. Step 3. and 4. can be repeated until a single cluster remain.

Multiple linkage criteria are available:

- *Average-linkage* : use the average score of each link merged in the cluster
- *Complete-linkage* : use the maximal score of each link merged in the cluster
- *Single-linkage* : use the minimal score of each link merged in the cluster

Example 7 shows an example for the merging procedures and linkage criteria.

The merging procedure can be stopped with one of the two following criteria:

- When a certain cluster number is reached. This method is not really helpful for our problem, since we assumed that we do not know the number of authors in the corpus.
- When the lowest score for the next merge is above a certain value. This value is called the distance threshold.

The so-called cut can be associated to position where the procedure is stopped. The cut is a theoretical position in the rank list which maximize true links above the cut and false link under. In the other hand, the stop position is the position at which the algorithm should stop merging clusters. They are not exactly the same, since when clusters are merged in the distance matrix, the resulting distance matrix can not be expressed as a rank list anymore.

The two supervised methods proposed in this study try to estimate the distance threshold, by estimating the optimal cut. Once a distance threshold is estimated, the hierarchical clustering is run on the rank list and is stopped according to the distance threshold.

There is one main flaw with the current scikit-learn implementation. It does not provide a way to access the clusters at each merging step. This feature is required for the unsupervised method, which evaluated multiple clustering results.

A possible workaround is to run the algorithm multiple time. Each time, the algorithm is stopped at a different number of clusters (a different step). Though this workaround is valid, it still introduces unnecessary additional merges. This overhead have a complexity of $O(1+2+3+...+N-1) = O(\frac{N*(N-1)}{2}) = O(N^2)$, with N equal to the number of documents. In other words, with the workaround, each time the algorithm is run, all the merges computed at the previous run must re-recomputed. To avoid this overhead, the agglomerative clustering algorithm was reimplemented to allow access to the clusters at each step.

5.1.2 Parameters

As explained in Section 5.1.1, the hierarchical clustering decide clusters based on a bottom-up approach and have two main parameters, the stopping procedure and the linkage criterion. The linkage criterion can be: single, average or complete linkage. Oxquarry, Brunet, St-Jean A, St-Jean B and St-Jean corpora are used for this experiment.

The goal is to understand how the linkage criterion behave for the best clustering achievable (best $B^3_{F_1}$).

The custom implementation of the hierarchical clustering is used, at each merging step, the intermediary clustering is evaluated using the

Example 7: Agglomerative clustering

(a) Initial clusters and their distances

	A	B	C	D
A	-	**1**	2	3
B	-	-	8	7
C	-	-	-	6
D	-	-	-	-

The link with the smallest distance is A-B with a distance of 1. At the next step, the clusters A and B are merged into a cluster called AB. (b), (c) and (d) shows how to update the distance matrix in (a) depending on the linkage criterion.

(b) First merge using single-linkage

	AB	C	D
AB	-	$\min{[2,8]} = 2$	$\min{[3,7]} = 3$
C	-	-	6
D	-	-	-

Next step will merge AB and C.

(c) First merge using average-linkage

	AB	C	D
AB	-	$\text{avg}\,[2,8] = 5$	$\text{avg}\,[3,7] = 4$
C	-	-	6
D	-	-	-

Next step will merge AB and D.

(d) First merge using complete-linkage

	AB	C	D
AB	-	$\max{[2,8]} = 8$	$\max{[3,7]} = 7$
C	-	-	6
D	-	-	-

Next step will merge C and D.

$B^3_{F_1}$. The clustering resulting in the best $B^3_{F_1}$ is kept. For each linkage criteria, corpus and text representation clustering and its evaluation is achieved.

The results are presented in Table 9.

These results can be considered as the $B^3_{F_1}$ upper bound for the hierarchical clustering using these rank lists on these corpora. We refer to them as *upper bound* or *rank lists upper bound.*

The best linkage criterion depends on the corpus, but in average, the best $B^3_{F_1}$ is achieved with the average linkage criterion. The average linkage is $\sim 1\%$ better than the complete linkage and $\sim 6\%$ better than the single linkage.

We assumed that the better a rank list is, the better the clustering result is when applying the hierarchical clustering. This assumption can be verified in Figure 13. In this figure, we compare the average precision (AP) of each retained rank list to the $B^3_{F_1}$ of the best clustering obtainable with the hierarchical clustering on this rank list (ref. Table 9).

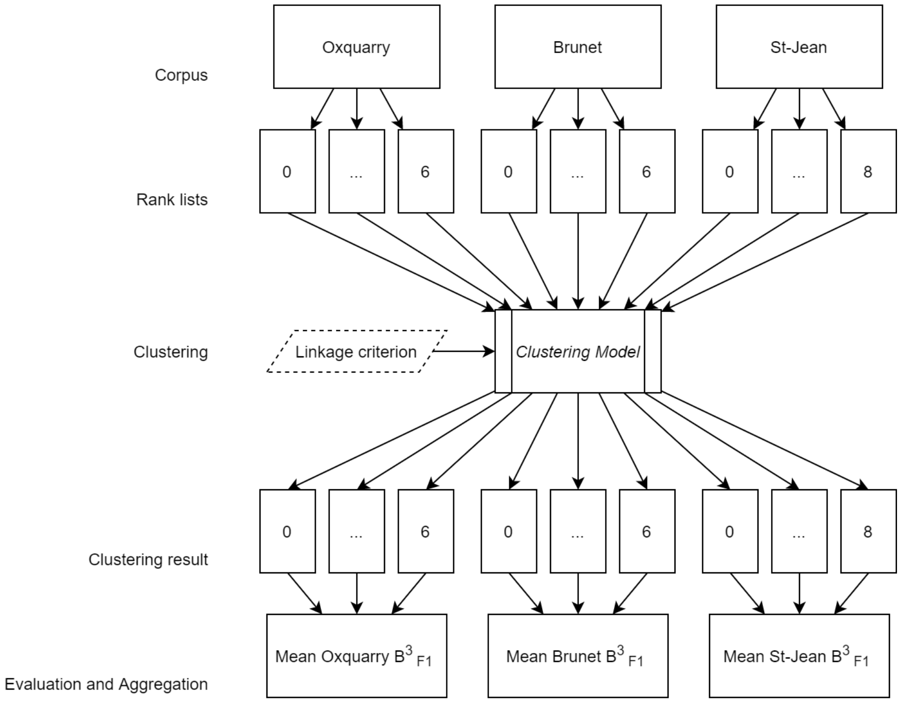
We can clearly observe a linear relationship between the average precision and the $B^3_{F_1}$. The linear regression for the single linkage have an r-value of 10^{-11}, 10^{-12} for the average linkage and 10^{-6} for the complete linkage. This result further motivate this assumption.

5.2 Silhouette-based clustering

5.2.1 Method

The main idea of the Silhouette-based clustering is to evaluate in an unsupervised manner the clustering result for each number of clusters (at each merge step of the hierarchical clustering). When discarding the clustering with N and clusters 1, this produce $N - 2$ possible clustering, each of those are evaluated using the unsupervised mean Silhouette score metric. The mean Silhouette score is defined in Definition 38.

Figure 12: Evaluation methodology for the clustering



Definition 38 - Mean Silhouette score [34] [35]
The mean Silhouette score s is an unsupervised clustering metric which evaluate a clustering result by measuring the cohesion $a(i)$ and separation of the clusters $b(i)$.

$$s = \frac{1}{|C|} \sum_{i=0}^{|C|} \frac{b(i) - a(i)}{max(a(i), b(i))}$$

$a(i)$: mean intra-cluster distance

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

$b(i)$: mean nearest-cluster distance

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

With C the set of clusters, C_i the i-th cluster, $d(i,j)$ the distance between the document i and j. $d(i,j)$ are pre-computed and also appear in the rank list.
The value is ranged between -1 and 1, a large value indicate a good cohesion and good separation of the clusters (low intra-cluster distance, high nearest-cluster distance).

The right number of cluster is not known. Using this technique, we suppose that the best number of clusters, is the one with the largest mean Silhouette score. Each step of the hierarchical clustering is evaluated with the mean Silhouette score. Only the one with the largest is kept.

An alternative to this method is the Iterative Positive Silhouette (IPS) and was proposed in Layton, Watters, Dazeley (2011) [4]. This method uses the median Silhouette score instead, and use another stopping procedure based on the sign of the score.

5.2.2 Evaluation

For this experiment, the goal is to test the Silhouette-based hierarchical clustering method on the literature corpora.

The rank lists used for this experiment are the ones from the retained text representation (9 for St-Jean and 7 for Brunet and Oxquarry). These are presented in Annex (Table K and Table L).

The evaluation methodology is presented in the schema in Figure 12. The $B^3_{F_1}$ score and r_{diff} average on the Silhouette-based clustering are presented in Table 10. The complete table is available in Annex (Table M).

The average r-ratio difference is a positive value ranging between $[0.17, 0.22]$ depending on the linkage criterion. Having an r-ratio larger than 0 indicate that the estimated number of cluster on every corpus is overestimated. This means that

Table 9: Best $B^3_{F_1}$ with hierarchical clustering on every rank lists and linkage criterion

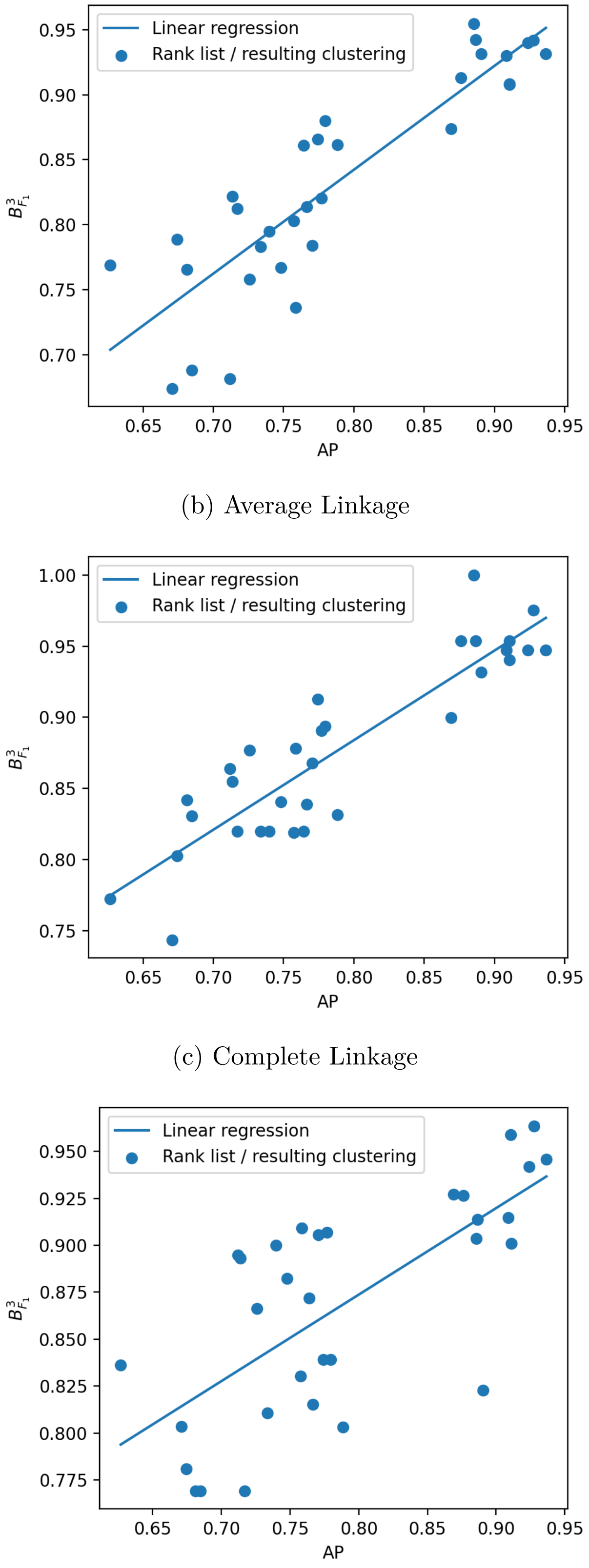
Rank list		Linkage criterion		
TR ID	Corpus	Single	Average	Complete
0	Oxquarry	0.95	1.00	0.90
1	Oxquarry	0.93	0.93	0.82
2	Oxquarry	0.67	0.74	0.80
3	Oxquarry	0.77	0.77	0.84
4	Oxquarry	0.87	0.91	0.84
5	Oxquarry	0.88	0.89	0.84
6	Oxquarry	0.86	0.83	0.80
Oxquarry mean		0.85	0.87	0.84
0	Brunet	0.81	0.82	0.77
1	Brunet	0.82	0.85	0.89
2	Brunet	0.77	0.84	0.77
3	Brunet	0.79	0.80	0.78
4	Brunet	0.78	0.82	0.81
5	Brunet	0.79	0.82	0.90
6	Brunet	0.86	0.82	0.87
Brunet mean		0.80	0.83	0.83
0	St-Jean A	0.82	0.89	0.91
1	St-Jean A	0.77	0.84	0.88
2	St-Jean A	0.78	0.87	0.91
3	St-Jean A	0.80	0.82	0.83
4	St-Jean A	0.68	0.86	0.89
5	St-Jean A	0.74	0.88	0.91
6	St-Jean A	0.76	0.88	0.87
7	St-Jean A	0.69	0.83	0.77
8	St-Jean A	0.81	0.84	0.82
St-Jean A mean		0.76	0.86	0.86
0	St-Jean B	0.93	0.95	0.95
1	St-Jean B	0.91	0.94	0.96
2	St-Jean B	0.94	0.98	0.96
3	St-Jean B	0.94	0.95	0.91
4	St-Jean B	0.93	0.95	0.91
5	St-Jean B	0.94	0.95	0.94
6	St-Jean B	0.87	0.90	0.93
7	St-Jean B	0.91	0.95	0.90
8	St-Jean B	0.91	0.95	0.93
St-Jean B mean		0.92	0.95	0.93
Absolute mean		0.83	0.88	0.87

the mean neareast-cluster distance is greater than the mean intra-cluster distance, even when dealing with the right number of clusters. In Table 10, the r-ratio is clearly too high, and the $B^3_{F_1}$ score is $\sim 14\%$ worse than the upper bound.

This can be due to the fact that the rank list used for the clustering is not perfect (AP $\neq 1$).

Figure 13: Optimal clustering $B^3_{F_1}$ correlation with rank list AP

(a) Single Linkage



(b) Average Linkage

(c) Complete Linkage

Table 10: Silhouette-based clustering evaluation (Maximal Silhouette, $\alpha = 0$), mean $B^3_{F_1}/r_{diff}$

	Linkage criterion		
Corpus	Single	Average	Complete
Oxquarry	0.76/0.18	**0.79/0.12**	0.78/0.13
Brunet	0.69/0.28	0.71/0.26	**0.73/0.23**
St-Jean A	0.59/0.33	**0.64/0.24**	0.61/0.26
St-Jean B	**0.91/0.08**	**0.91/0.06**	0.90/0.06
Absolute mean	0.74/0.22	**0.76/0.17**	0.75/0.17
Upper bound	*0.83/0.00*	*0.88/0.00*	*0.87/0.00*

In bold: For each corpus, the criterion with the largest $B^3_{F_1}$

Table 11: Silhouette-based clustering evaluation ($\alpha = -0.2$), mean $B^3_{F_1}/r_{diff}$

	Linkage criterion		
Corpus	Single	Average	Complete
Oxquarry	**0.81/0.06**	0.78/0.03	0.79/0.02
Brunet	0.78/0.10	**0.80/0.09**	**0.80/0.10**
St-Jean A	0.70/0.14	**0.77/0.08**	0.76/0.09
St-Jean B	0.86/0.02	**0.87/0.03**	0.86/0.03
Absolute mean	0.79/0.08	**0.81/0.06**	0.80/0.06
Upper bound	*0.83/0.00*	*0.88/0.00*	*0.87/0.00*

In bold: For each corpus, the criterion with the largest $B^3_{F_1}$

5.2.3 Tweak for Authorship Clustering

We want to mitigate having the number of cluster overestimated by the Silhouette-based clustering method. An easy solution is to use the clustering results produced earlier in the algorithm steps (less merges, less clusters, less overhestimation of the number of clusters). This corresponds to a non-maximal value of the mean Silhouette score, on the left side of the maximal mean Silhouette score. This procedure is a form of early stop.

In this study, we introduce a parameter called α. α represents a percentage to subtract to the maximal mean Silhouette score to obtain a new target with a lower mean Silhouette score (instead of the maximal value).

The mean Silhouette score across the number of clusters is a concave function. The new target, can thus be either on the left or on the right of the maximal mean Silhouette. To remove this ambiguity, the sign of α indicate on which side of the maximal Silhouette score the target should be.

With $\alpha = 0$, this corresponds to the maximal mean Silhouette score. With a negative α the left side is targeted and with a positive α, the right side.

The clustering result with the Silhouette score the closest to the target on the desired side is retained.

Definition 39 - α-Silhouette
The α-Silhouette target score found by using the maximal Silhouette score as a baseline and is adjusted using a parameter called α.

$$\begin{aligned} target &= max(Scores) - |\alpha| \cdot max(Scores) \\ &= max(Scores) \cdot (1 - |\alpha|) \end{aligned}$$

Notice, the sign of the α parameter is not taken into account to select the score target.

Example 8 shows the α computation and usage.

For example, by using $\alpha = -0.2$, we aim to correct this overshoot and increase the quality of the clustering results. This value was chosen by grid search to optimize the $B^3_{F_1}$.

Table 11 show the evaluation with $\alpha = -0.2$, the complete table is available in Annex (Table N).

With this correction technique, in average the r_{diff} is closer to 0 and the average $B^3_{F_1}$ is increased in average by 7% across all the corpora. With $\alpha = -0.2$, we obtained a $B^3_{F_1}$ only 8% worse than the upper bound, instead of the 14% obtained with $\alpha = 0$. The average linkage criterion give the best results for this clustering method. The $B^3_{F_1}$ for the average linkage criterion is in average 2% better than the other criterions.

5.3 Distribution-based clustering

5.3.1 Method

In Savoy (2014)'s *Estimating the Probability of an Authorship Attribution* [7] they modelled the true

Example 8: α correction

(a) Silhouette Scores for each number of clusters

Number of clusters	Silhouette Score
3	2.5
4	3.2
5	3.5
6	3.9
7	3.1
8	2.9

(b) α computations

with $max(Scores) = 3.9$ and $\alpha = -0.2$

$$\begin{aligned} target &= 3.9 \cdot (1 - |-0.2|) \\ &= 3.9 \cdot 0.8 \\ &= 3.12 \end{aligned}$$

(c) Clustering selection

Since α is negative, the left side of the maximal mean Silhouette is used (negative: smaller number of clusters).
Thus, the clustering on the left side with the closest Silhouette score to the target is the one with 4 clusters (score: 3.2).
With $\alpha = 0.2$, the target would be the same, but the right side would be selected instead. The number of cluster selected would be 7.
With $\alpha = 0$, the number of cluster selected would be 6, since the target would be 3.9.

and false links score distribution using two Beta distribution.

As explained in [7] the Beta distribution is better suited for authorship problem than, for example, the Gaussian distribution since it can grasp a larger amount of distribution shapes with its parameters flexibility.

In this study, we use these two models to find the position where the sum of the two areas under the curve is maximized. It corresponds to the position in the rank list where the links true positive and true negatives are maximized according to the model (in a non-weighted minimization errors schema). This position also correspond to where the two models have the same area under the curve and is called the equiprobable position.

Each Beta distribution require two parameters: a and b. Firstly to compute the parameters, the population must be normalized between 0 and 1 using Definition 40. The population are represented, for each distribution, by the true links or the false links scores.

Definition 40 - Linear normalization
Normalize a vector X in the interval $[0, 1]$.

$$\begin{aligned} \alpha &= Min(X) \\ \beta &= Max(X) - Min(X) \\ Norm(X) &= \frac{X - \alpha}{\beta} \end{aligned}$$

Denormalize a scalar or a vector X from the interval $[0, 1]$ to the interval $[\alpha, \alpha + \beta]$

$$DeNorm(X, \alpha, \beta) = X \cdot \beta + \alpha$$

The a and b parameters are estimated using the distribution population mean and the variance, see Definition 41

Definition 41 - Beta distribution parameters [7]

$$\begin{aligned} \mu &: \text{The population mean} \\ \sigma &: \text{The population standard deviation} \end{aligned}$$

$$\begin{aligned} \Delta &= \frac{\mu \cdot (1 - \mu)}{\sigma} - 1 \\ a &= \mu \cdot \Delta \\ b &= (1 - \mu) \cdot \Delta \end{aligned}$$

The Beta distribution probability density function (PDF) is computed using the formula in Definition 42

Definition 42 - Beta distribution PDF [7]

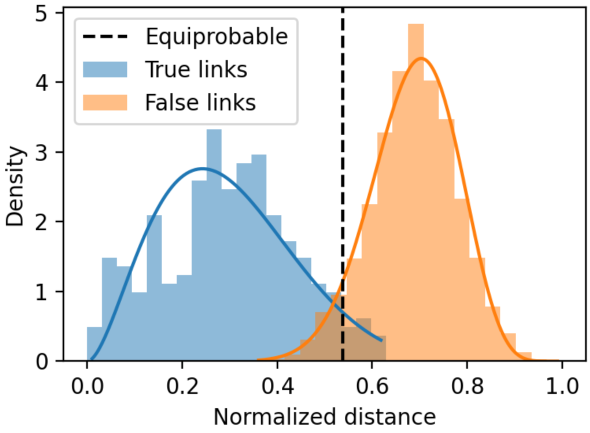
$$Beta(X|a, b) = \frac{\Gamma(a + b)}{\Gamma(a) \cdot \Gamma(b)} \cdot X^{a-1} \cdot (1 - X)^{b-1}$$

with $a > 0$, $b > 0$ and $\Gamma()$ the Gamma function

The equiprobable position is found using a binary search between 0.00 and 1.00. At each step, the cumulative distribution function (CDF) of the two Beta distribution are evaluated. From left to right for the distribution on the left, and from right to left for the distribution on the right. To compute this function, the scipy library is used.

Once the two area are obtained, their difference is computed. The binary search stops once the difference is a small value, such as 10^{-15}. Binary searches have a complexity of $O(\log n)$.

Figure 14: Links distances density and Beta distribution estimation for St-Jean B with text representation 0



There might be analytical ways to solve this problematic using the *Beta distribution cumulative distribution function analytic form* (Beta CDF analytic form) but was out of the scope of this study. With an analytical solve, the complexity could drop to $O(1)$.

Figure 14 shows the distance density for true and false link as well as a Beta distribution estimation for the St-Jean B rank list generated with text representation 0 (ref. Section 4.6).

The vertical line in this figure, indicates the equiprobable position where both Beta distribution have the same probability of being a true link and false link (same area under the curve). This point can be used as a decision point where the cut should be made in the rank list, this ensures that both false positives and false negatives are minimized.

Once the equiprobable position is found for a corpus with known authors, it can be re-used for new corpora. After denormalizing (see Definition 40), the equiprobable position is used as a distance threshold to stop the hierarchical clustering.

This method is supervised, since it requires examples to learn the cut. The distribution-based clustering do not consider any new corpus for its value computation. Each training rank list yields a single fixed real number, the optimal cut position for this rank list.

5.3.2 Possible Cost Function

This technique can be adapted, in the case where a loss/cost function have to be minimized. This can be used when the false positives do not have the same importance as false negatives.

To introduce a cost function to the two Beta method, the position search criterion is changed. Instead of finding the position where both the true link and false link area correspond to 50% of their sum (equiprobable case). We can aim to find where true links area correspond to, for example, 60% and false links area to 40% in this case.

It can be generalized such that the true link area represent α and the false link area to $1 - \alpha$ with $\alpha \in [0, 1]$. When α is greater than 0.50, more false positives and less false negatives will occur. When α is smaller than 0.50, less false positives and more false negatives will occur.

For the clustering, the α parameter directly influence the r_diff value and the $B^3_{precision}$ and B^3_{recall}. The same binary search can be used for this computation, just the target need to be changed. In this study, we only considered the equiprobable case for the experiments.

5.3.3 Evaluation

The distribution-based clustering approach is evaluated using the Oxquarry, Brunet, St-Jean A and B corpora. The retained rank list for each corpus used are in Annex (Table K and Table L). For each rank list, the distance threshold is computed using the two Beta approach. This step corresponds to the training phase.

Then for every distance threshold and rank list pair, the hierarchical clustering is applied. This corresponds to the testing phase. The results are evaluated using the $B^3_{F_1}$ and the r_{diff} metrics. The evaluation results are aggregated by applying the arithmetic mean on the $B^3_{F_1}$ and r_{diff} across all retained rank list.

The evaluation is presented in Table 12 which show the results for each linkage criterion (Single, Average, Complete).

The following conclusions can be drawn with these results:

- In terms of linkage criteria, the single linkage is not adapted, with an average $B^3_{F_1} = 0.22$.
- Average linkage have the best $r_{diff} = 0.06$ which indicate that this criterion can estimate the most accurately the number of clusters.

- Complete linkage have the best $B^3_{F_1} = 0.82$ which make it the best criterion for this method. Which is 8% better than the average linkage and 273% better than the single linkage.
- The training corpus does not influence a lot the quality of the results.

  $std(\text{MeanTraining}_{B^3_{F_1}}) = 0.01$

  The best corpus to train is Brunet, with slightly better results.
- In the other hand, the quality of the rank list used for the testing phase is more impactful.

  $std(\text{MeanTesting}_{B^3_{F_1}}) = 0.04$

  This further validate the assumption made in Section 5.1.2. St-Jean B's rank list have the best average precision and the best clustering results.

5.4 Regression-based clustering

5.4.1 Method

To learn at which position in the rank list the cut should be, this third idea is to train a linear model. The model aims to learn to discriminate the same authors document pairs from different authors document pairs in a rank list.

To train the model, samples are created for each link in a training rank list. The links labels are either *true links* when both document in the link are from the same author or *false link* otherwise.

The two features used are: the log of the link relative rank ($log\frac{link\_rank}{|L|}$) and the score of the link.

First feature provide to the model that true links are generally at the top of the rank list. The value is normalized by the size of the rank list. This allows to train in more generically manner, such that the model can be used to predict on any rank list size. The logarithm scale is used to have a greater granularity for the top ranked links.

Second feature aims to provide to the model that the small distances or large similarities are generally true links. Using these two features, the model can, depending on each link rank and the score, determinate if the link is a true link. A model trained with these features does not depend on the text language nor the corpus size. This model is metric dependent, since the score magnitude change according to the distance function.

In this study, the linear model used is the logistic regression. The advantage of using a regression model is that the output of the model will correspond to a probability of being a true link. To find the cut in the testing rank list, the fitted model predict the probability of being a true link for each link. A trained logistic regression model can be used to predict true link probabilities on any other rank lists produced with the same distance metrics.

From these predictions, a probability threshold must be chosen. For example, having a probability threshold at 0.5, aims to minimize both false negatives and the false positives. The threshold used to optimize the $B^3_{F_1}$ should be 0.5, since this metrics aims to optimize both the recall and the precision which respectively aims to minimize false negatives and false positives.

The probability threshold can be adjusted for a cost minimization case. For example if false negatives are more important to minimize, a probability threshold at 0.6 can be selected instead or 0.4 if the false positive should be minimized. Which will subsequently either improve the B^3_{recall} or the $B^3_{precision}$ for the authorship clustering.

The distance threshold is selected according to the score of the closest link to the probability threshold. With this method, the distance threshold can be imprecise. This is caused by the fact that the rank list is in a discrete space and the score in a continuous space. In other words, there might not be a link with a true link probability of 0.5 and the distance threshold will not be selected properly with this method.

One way to find a slightly more accurate distance threshold is to use a linear interpolation. The score is interpolated to the true links probabilities the closest to the threshold, respectively, the one directly above and the one directly below in the rank list. Example 9 showcases a linear interpolation of the score using a 0.5 probability threshold in a synthetic rank list.

Table 12: Distribution-based clustering evaluation, mean $B^3_{F_1}/r_{diff}$ for each corpus pair

(a) Single Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.27/0.15	0.18/0.22	0.14/0.16	0.12/0.18	0.18/0.18
	Brunet	0.36/0.12	0.18/0.22	0.15/0.16	0.12/0.18	0.20/0.17
	St-Jean A	0.42/0.11	0.30/0.19	0.14/0.16	0.13/0.18	0.25/0.16
	St-Jean B	0.42/0.10	0.35/0.16	0.16/0.16	0.16/0.17	0.27/0.15
	Mean	0.37/0.12	0.25/0.20	0.15/0.16	0.13/0.18	0.22/0.16

(b) Average Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.73/0.04	0.65/0.08	0.59/0.10	0.61/0.11	0.65/0.08
	Brunet	0.78/0.05	0.73/0.04	0.73/0.07	0.73/0.08	0.74/0.06
	St-Jean A	0.82/0.05	0.74/0.06	0.81/0.04	0.83/0.05	0.80/0.05
	St-Jean B	0.83/0.09	0.79/0.08	0.83/0.02	0.90/0.02	0.84/0.05
	Mean	0.79/0.06	0.73/0.06	0.74/0.06	0.77/0.06	0.76/0.06

(c) Complete Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.79/0.06	0.77/0.05	0.82/0.04	0.84/0.04	0.81/0.05
	Brunet	0.79/0.10	0.81/0.09	0.84/0.02	0.90/0.02	0.83/0.06
	St-Jean A	0.79/0.11	0.81/0.11	0.82/0.05	0.90/0.02	0.83/0.07
	St-Jean B	0.78/0.13	0.79/0.14	0.77/0.09	0.90/0.04	0.81/0.10
	Mean	0.79/0.10	0.80/0.10	0.81/0.05	0.88/0.03	0.82/0.07

5.4.2 Evaluation

To evaluate the regression-based clustering approach: the Oxquarry, Brunet, St-Jean A and B corpora were used.

Each retained rank list for each corpus is computed and used for the experiment, see their evaluation in Annex (Table K and Table L). A logistic regression model is trained with each rank list. This step corresponds to the training phase.

Then the trained models are used to predict a distance threshold for the rank list with the same distance metrics. The hierarchical clustering is applied to the rank list using the distance threshold predicted. The clustering result is evaluated using the $B^3_{F_1}$ and the r_{diff}. In other words, a model to find distance threshold is trained on every rank list from every corpus and tested on every rank list from every corpus. This corresponds to the testing phase.

The experiment is repeated for each linkage criterion.

The results are presented in Table 13. The results are aggregated using the arithmetic mean on the $B^3_{F_1}$ and r_{diff} on all retained rank list per corpus as shown in Figure 12 in Section 5.1.2.

The following conclusions can be drawn with these results:

- The best linkage criterion for this model is the average linkage with an average $B^3_{F_1} = 0.80$. The average linkage have a relative $B^3_{F_1}$ increase of 20% over the single linkage and 8% over the complete linkage.
- Some corpus such as the Oxquarry corpus have better rank list for training the clustering model even though their rank list have a

Example 9: Linear interpolation for regression-based clustering distance threshold selection (probability threshold fixed at 0.5)

(a) Rank list with link probability and score

Rank	Probability	Score
(...)		
45th	0.54	15
46th	0.52	13
47th	0.49	12
48th	0.48	10
(...)		

(b) Linear interpolation

$$\alpha = \frac{0.5 - 0.49}{0.52 - 0.49} = \frac{1}{3}$$

$$distance\_threshold_{@0.5} = (13 - 12) \cdot \alpha + 12 = 12.\bar{3}$$

worse average precision than other corpora. When training with Oxquarry corpus with the average linkage, the $B^3_{F_1}$ is 9% better than the ones trained with the Brunet corpus. This indicates that the training set does not necessarily require a high quality rank list to train the clustering model.

- When testing the cut model, having a rank list of good quality tends to produce a better clustering, no matter the quality of the rank list used for the training. As shown in Section 5.1.2. For example, when testing the clustering model on the corpus with the best rank list in average (St-Jean B), it obtains in average the best clustering, $B^3_{F_1} = 0.91$.

The conclusion to the previous points is that having a good rank list is more important for testing than training. Though some corpus can be better than others for training. No clear rank lists property was found, which makes them better to train the model over others.

5.5 Clustering Summary

Three methods based on the hierarchical clustering are proposed to solve the authorship clustering task.

The first is based on the mean Silhouette score to find the best clustering at each step of the hierarchical clustering. The second uses an author links distribution model to find the best step to stop the hierarchical clustering. And the last is based on a logistic regression model to estimate the true link probability for new rank lists. These methods are summarized in the schema in Figure 15.

Table 14 contains a evaluation summary of the proposed method. The evaluation use the $B^3_{F_1}$ score and the r_{diff} metrics.

Each method produce similar results, but the distribution-based clustering with complete linkage give the best results, $B^3_{F_1} = 0.82$. The Silhouette-based methods and the regression-based method produce slightly less accurate results. The non-tweak Silhouette-based model give the least accurate results.

The complete linkage and average linkage criterion give the best results. Though the average linkage criterion have better results with three out of the four models, the distribution based model with the complete linkage give the best results across the board. It reaches an $B^3_{F_1} = 0.82$. Since the single linkage criterion produce weak results for every model, this criterion is left aside for the next experiments.

Three models give the best approximation of the number of clusters with $r_{diff} = 0.06$: The Silhouette-based model with average linkage or complete linkage, and the distribution-based model with average linkage. These models are close to find the right number of clusters almost every time.

We compare the models with the upper bound. The upper bound is the best clustering achievable using the same rank lists as for the model evaluation. The upper bound is found by evaluating the clusters $B^3_{F_1}$ score at each step of the hierarchical clustering. The step with the best metrics are kept for each rank list. The values exposed in this table are the average metrics of each *best clustering* for the rank lists.

Overall, the results are close to the upper bound. The best $B^3_{F_1}$ achieved with each model is $8 - 15\%$ worse than the upper bound.

Table 13: Regression-based clustering evaluation, Mean retained rank lists $B^3_{F_1}/r_{diff}$ for each corpus pair

(a) Single Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.58/0.09	0.59/0.09	0.29/0.14	0.22/0.16	0.42/0.12
	Brunet	0.77/0.17	0.78/0.09	0.58/0.10	0.83/0.03	0.74/0.10
	St-Jean A	0.79/0.14	0.77/0.06	0.50/0.09	0.69/0.07	0.69/0.09
	St-Jean B	0.78/0.15	0.78/0.08	0.53/0.08	0.74/0.05	0.71/0.09
	Mean	0.73/0.14	0.73/0.08	0.48/0.10	0.62/0.08	0.64/0.10

(b) Average Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.83/0.11	0.80/0.11	0.83/0.04	0.91/0.01	0.84/0.07
	Brunet	0.72/0.22	0.75/0.20	0.73/0.15	0.89/0.07	0.77/0.16
	St-Jean A	0.73/0.21	0.76/0.17	0.76/0.13	0.92/0.05	0.79/0.14
	St-Jean B	0.72/0.21	0.75/0.19	0.75/0.13	0.92/0.05	0.79/0.15
	Mean	0.75 0.19	0.76 0.17	0.77 0.11	0.91 0.05	0.80/0.13

(c) Complete Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.76/0.15	0.79/0.16	0.72/0.14	0.91/0.04	0.79/0.12
	Brunet	0.63/0.30	0.71/0.24	0.65/0.21	0.82/0.13	0.71/0.22
	St-Jean A	0.66/0.27	0.73/0.23	0.67/0.19	0.85/0.10	0.73/0.20
	St-Jean B	0.64/0.29	0.73/0.23	0.67/0.20	0.84/0.10	0.72/0.20
	Mean	0.67/0.25	0.74/0.22	0.68/0.19	0.86/0.09	0.74/0.19

Figure 15: Clustering methods schema

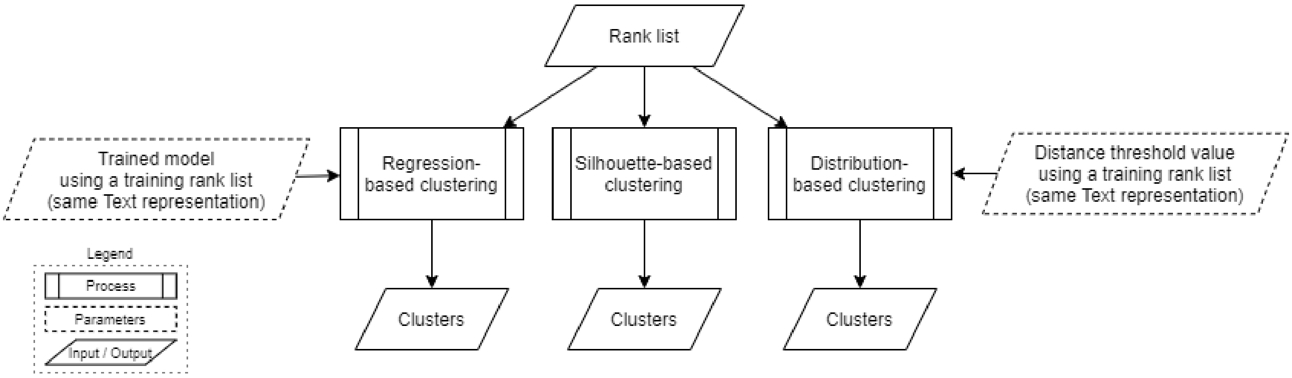

Rank list
Trained model using a training rank list (same Text representation)
Regression-based clustering
Silhouette-based clustering
Distribution-based clustering
Distance threshold value using a training rank list (same Text representation)
Clusters
Clusters
Clusters
Legend
Process
Parameters
Input / Output

Table 14: Mean retained rank lists $B^3_{F_1}/r_{diff}$ for each clustering method

	Linkage criterion		
Clustering method	Single	Average	Complete
Silhouette-based ($\alpha = 0$)	0.74/0.22	**0.76/0.17**	0.75/0.17
Silhouette-based ($\alpha = -0.2$)	0.79/0.08	**0.81/0.06**	0.80/0.06
Distribution-based	0.22/0.16	0.76/0.06	**0.82/0.07**
Regression-based	0.64/0.10	**0.80/0.13**	0.74/0.19
Rank lists upper bound	*0.83/0.00*	*0.88/0.00*	*0.87/0.00*

In bold: The linkage criterion with the largest $B^3_{F_1}$ for each clustering method.

Chapter 6

Rank List Fusion

To increase the rank list quality, the proposed method is to use a combination of multiple rank list to form an optimistically better rank list. We call this method *fusion* or *rank list fusion*. By using the most diverse strategies, we believe it is possible to increase the rank list quality using the principle of combination of evidences.

In this chapter, two fusion strategies will be presented. To evaluate the two fusion schemes, techniques from Section 3.6.4 are used.

At the end of this chapter, the rank list produced by fusion will be used to solve the authorship clustering task.

6.1 Methods

The idea of rank list fusion is to combine multiple rank lists in such a way that the resulting rank list is a more accurate modelling of a perfect rank list. A perfect rank list contain every true links (same author document pairs) at the top and every false links (different author document pair) at the bottom, therefore maximizing the metrics in Section 3.6.2.

The first proposed method is based on the Z-Score normalization. The second one use the logistic regression.

In this thesis, we assume that each rank for the fusion have the same importance. Thus, the fusions are based on a non-weighted arithmetic mean.

6.1.1 Z-Score Fusion

When merging rank lists, the following problem arise: Due to the difference of distance metric and/or text representation, the scores used in the different rank lists are of different order of magnitude. Simply applying the arithmetic mean on the scores for each rank list is thus not effective.

To mitigate this problem, a simple and rather effective approach is to normalize the scores of all rank lists independently using the Z-Score (See Definition 28 in Section 4.1.1). Then one can compute a resulting score for each link as the arithmetic mean of scores in every Z-Score normalized rank lists. The rank list can then be sorted by this new score.

Additionally, this method provide a framework for the fusion with both distance metrics and similarity metrics. If distances and similarity score are encountered for a fusion, flipping the score sign for either the distance rank lists or similarity rank lists before or after the Z-Score normalization allow to merge them.

Even though this method lack theoretical foundations, it can provide good results and does not have any parameters.

Example 10 exemplifies the Z-Score fusion computation using two simple rank lists.

6.1.2 Regression Fusion

This second fusion method is based on logistic regression and is called regression fusion in this study.

Each rank list is ordered by a score, this score depends on the distance measure and text representation used, thus the order of magnitude across the rank list can be different. To avoid this problem, the second solution is to express the scores into a probability of being a true link acording to a model. To do so, the logistic regression is used. This method is similar to the one proposed in *Le Calvé and Savoy's (2000)* paper [1] and was used to merge rank list issued from information retrieval systems.

Example 10: A two rank lists Z-Score fusion

(a) Rank list A (Mean = 50, Std = 50)

Rank	Link	Score	Z-Score
1st	(0, 2)	0	$\frac{(0-50)}{40.82} = -1.22$
2nd	(1, 2)	50	$\frac{(50-50)}{40.82} = 0$
3rd	(0, 1)	100	$\frac{(100-50)}{40.82} = 1.22$

In this example, for the sake of simplicity, the scores/distances does not satisfy the triangle inequality.

(b) Rank list B (Mean = 0.4, Std = 0.435)

Rank	Link	Score	Z-Score
1st	(0, 2)	0.1	$\frac{(0.1-0.4)}{0.435} = -0.69$
2nd	(0, 1)	0.2	$\frac{(0.2-0.4)}{0.435} = -0.46$
3rd	(1, 2)	0.9	$\frac{(0.9-0.4)}{0.435} = 1.15$

(c) Z-Score fusion

Rank	Link	Mean Z-Score
1st	(0, 2)	$\frac{-1.22+(-0.69)}{2} = -0.96$
2rd	(0, 1)	$\frac{1.22+(-0.46)}{2} = 0.38$
3nd	(1, 2)	$\frac{0+(1.15)}{2} = 0.575$

The idea is to learn a logistic regression model for each combination of text representation and distance metrics (denoted *type of rank list*). From each distinct rank list, features for each link are created. The features are: the log of the relative rank and the link score. Since a probability of being a true link is desired, the target value (labels) for each link is either 1.0 for the true links or 0.0 when it is a false link. This feature construction is the same as the one used for the regression-based clustering method (ref. Section 5.4).

One model per type of rank list is trained. This model can provide, for any new same type rank lists, true link probabilities for each link.

The probability framework can be used on the rank lists with probability scores. To obtain the most reliable probability given multiple observations (multiple rank lists), the regression fusion consists in computing the expected value of each link probability using these observations. In this case, the final value for sorting the rank list is an arithmetic mean of the probabilities, since no weighting is used on the rank lists.

The main advantage of this methodology is that it uses concrete statistical concepts. But this technique has one main issue: it requires training samples to train a model.

6.1.3 Veto

To try to further improve the results, the idea here is to provide a veto power for the rank lists.

The following assumption is made: If a link have a low probability in an individual rank list (before the fusion), it probably indicates a false link. Having this link at the bottom of the rank list after the fusion should improve the results.

To fulfil this assumption, we introduce veto. Veto follows this algorithm:

1. In every rank lists just before the fusion in the regression (after converting into probabilities), find every link where the probability is under a certain threshold.
2. Alter their probabilities (probability of being a true link) with $-\infty$.
3. Perform the regression fusion (by averaging the probabilities)

No matter the rank list, every links below the probability threshold will have its final score also equal to $-\infty$. The average with any number and $-\infty$ is always $-\infty$. When sorting the rank list, the links with a resulting $-\infty$ score will subsequently be ranked at the bottom. This first veto method is called $-\infty$-veto in this study.

Example 12 showcases when the veto can provide better results than the standard fusion in Example 11. In the example, rank list 1 gives a 0.20 true link probability for a link with $ID = 4$, but rank list 3 give this link a 0.80 true link probability. The link with $ID = 3$ is ranked 3rd, but with the veto the link is ranked 5th.

One problem arise with this solution, multiple links can have a $-\infty$ probability. Thus, after the sort their position are random. In the case where false positives occurs (true links under the threshold), they will be randomly ranked at the bottom. This method is non-deterministic and non-idempotent.

To make them deterministic and idempotent, the proposed solution is to set the probability of the affected links to one of these values instead: 0, -1 or $-n$, with n equal to the number of rank lists to fuse. The greater, the value, the stronger the penalty for the links affected by the veto. The $-\infty$ method is the only real veto, in the first meaning of the veto definition, the other methods can be considered as a weighted veto.

The $-n$, method provide a way to avoid having tie and keep the strong impact of the $-\infty$-veto when a lot of rank list are fused. The n correspond to the number of rank lists to fuse. For example, if 9 rank lists are fused. By setting any link score to -9, this ensures that even if every other rank lists have a true link probability of 1.0 for this link. The resulting probability will be $(-9 + 8 \cdot 1.0)/9 = -0.11$, which is lower than 0.0, the lowest probability obtainable by the regression.

The theoretical maximal veto that keep the order is $nt + 1 - n - \epsilon$, with ϵ a small value (e.g. 2^{-20}) and t the threshold. Since even if all the other rank list have a probability of 1.0 with this veto, the values obtained after applying the veto are smaller than the threshold thus ranked at the rank list bottom.

Proof:

$$\frac{(nt + 1 - n - \epsilon) + (n-1) \cdot 1.0}{n} = \frac{nt - \epsilon}{n} = t - \frac{\epsilon}{n}$$

Instead $-n$ was chosen to keep this approach simple.

By setting to 0, -1 or $-n$, it lightens the veto effect and conserve an order for the bottom rank links. Example 13 show the 0-veto strategy. Here the link with $ID = 4$ is ranked 4th due to a lighter effect of this veto. Link with $ID = 5$ is still rank at the bottom.

6.1.4 Soft-veto

As an extension to the veto method, an additional desired constraint is to favor top ranked link and as well as penalizing bottom ranked links.

In any rank list, top ranked links should correspond to documents with the same author, and bottom ranked links should correspond to documents with different authors. Assuming that the top ranks are true links after the rank list fusion, these links should also appear top ranked. The same reasoning can be applied for the bottom links by assuming them as false links. A weighting curve can be design accordingly. The idea of this curve is to boost the score of top ranked links and hindered the score of bottom ranked links before the fusion.

Example 11: Fusion without veto

(a) Rank lists

Rank list 1		Rank list 2		Rank list 3	
Link ID	Score	Link ID	Score	Link ID	Score
0	0.95	0	0.90	5	0.90
1	0.75	3	0.70	4	0.80
2	0.60	4	0.65	0	0.70
3	0.50	1	0.50	1	0.60
4	0.20	2	0.30	2	0.50
5	0.10	5	0.10	3	0.40

(b) Rank lists fusion

Rank	Link ID	Average
1st	0	$(0.95 + 0.90 + 0.70)/3 = 0.85$
2nd	1	$(0.75 + 0.50 + 0.60)/3 = 0.62$
3rd	4	$(0.20 + 0.65 + 0.80)/3 = 0.55$
4th	3	$(0.50 + 0.70 + 0.40)/3 = 0.53$
5th	2	$(0.60 + 0.30 + 0.50)/3 = 0.47$
6th	5	$(0.10 + 0.10 + 0.90)/3 = 0.37$

Example 12: Fusion with $-\infty$-veto at 0.25

(a) Rank lists

Rank list 1		Rank list 2		Rank list 3	
Link ID	Score	Link ID	Score	Link ID	Score
0	0.95	0	0.90	5	0.90
1	0.75	3	0.70	4	0.80
2	0.60	4	0.65	0	0.70
3	0.50	1	0.50	1	0.60
4	-inf	2	0.30	2	0.50
5	-inf	5	-inf	3	0.40

(b) Rank lists fusion

Rank	Link ID	Average
1st	0	$(0.95 + 0.90 + 0.70)/3 = 0.85$
2nd	1	$(0.75 + 0.50 + 0.60)/3 = 0.62$
3rd	3	$(0.50 + 0.70 + 0.40)/3 = 0.53$
4th	2	$(0.60 + 0.30 + 0.50)/3 = 0.47$
5-6th	4	$(-\infty + 0.65 + 0.80)/3 = -\infty$
5-6th	5	$(-\infty + -\infty + 0.90)/3 = -\infty$

First, the score distribution must be understood. By observing the graph score/distance over rank, Figure 16, we can clearly observe the top rank links and bottom rank links have a sharper difference in distance than the ones in the middle section.

Example 13: Fusion with 0-veto at 0.25

(a) Rank lists

Rank list 1		Rank list 2		Rank list 3	
Link ID	Score	Link ID	Score	Link ID	Score
0	0.95	0	0.90	5	0.90
1	0.75	3	0.70	4	0.80
2	0.60	4	0.65	0	0.70
3	0.50	1	0.50	1	0.60
4	0.00	2	0.30	2	0.50
5	0.00	5	0.00	3	0.40

(b) Rank lists fusion

Rank	Link ID	Average
1st	0	$(0.95 + 0.90 + 0.70)/3 = 0.85$
2nd	1	$(0.75 + 0.50 + 0.60)/3 = 0.62$
3rd	3	$(0.50 + 0.70 + 0.40)/3 = 0.53$
4th	4	$(0.00 + 0.65 + 0.80)/3 = 0.48$
5th	2	$(0.60 + 0.30 + 0.50)/3 = 0.47$
6th	5	$(0.00 + 0.00 + 0.90)/3 = 0.30$

Using the inverse of the sigmoid function, a weighting curve satisfying our constraints can be modelled. The choice on the sigmoid function was arbitrary. Any function in the form of an S can be used, these are also called S-curves. The hyperbolic tangent or the arc tangent function are also possible functions.

Definition 43 show the sigmoid function and its inverse function.

Definition 43 - Sigmoid and Sigmoid inverse function
Sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}}$$

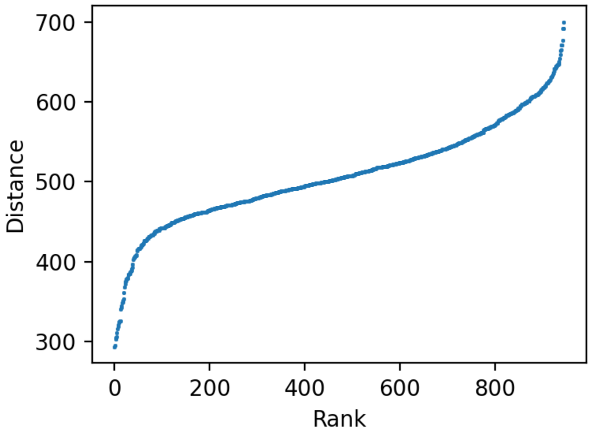
Sigmoid inverse:

$$S^{-1}(x) = -\ln \frac{x - 1}{x}$$

Since the sigmoid inverse function is continuous and the number of links in a rank list is discrete, the idea is to sample values in this curve at regular intervals. Also, since the sigmoid function is defined from $-\infty$ to $+\infty$ and the rank list have a finite size, an interval must be select.

This interval directly control the steepness of the curve and thus influence the strength of the veto. To control the steepness, a parameter called c is introduced. c correspond to a position on the x-axis of the sigmoid function. The sampling of the sigmoid inverse is done in the interval $[S(-c), S(c)]$. A large interval gives a greater steepness and a small interval a less steep curve. c can not be equal to 0 otherwise the interval consist of a single number (also known as degenerate interval). The c parameter must be in the interval $]0, \infty[$. With a value close to 0, the soft-veto have a weak effect. With a large value, for example 5, the soft veto has a strong effect on the top and bottom, see Figure 17a.

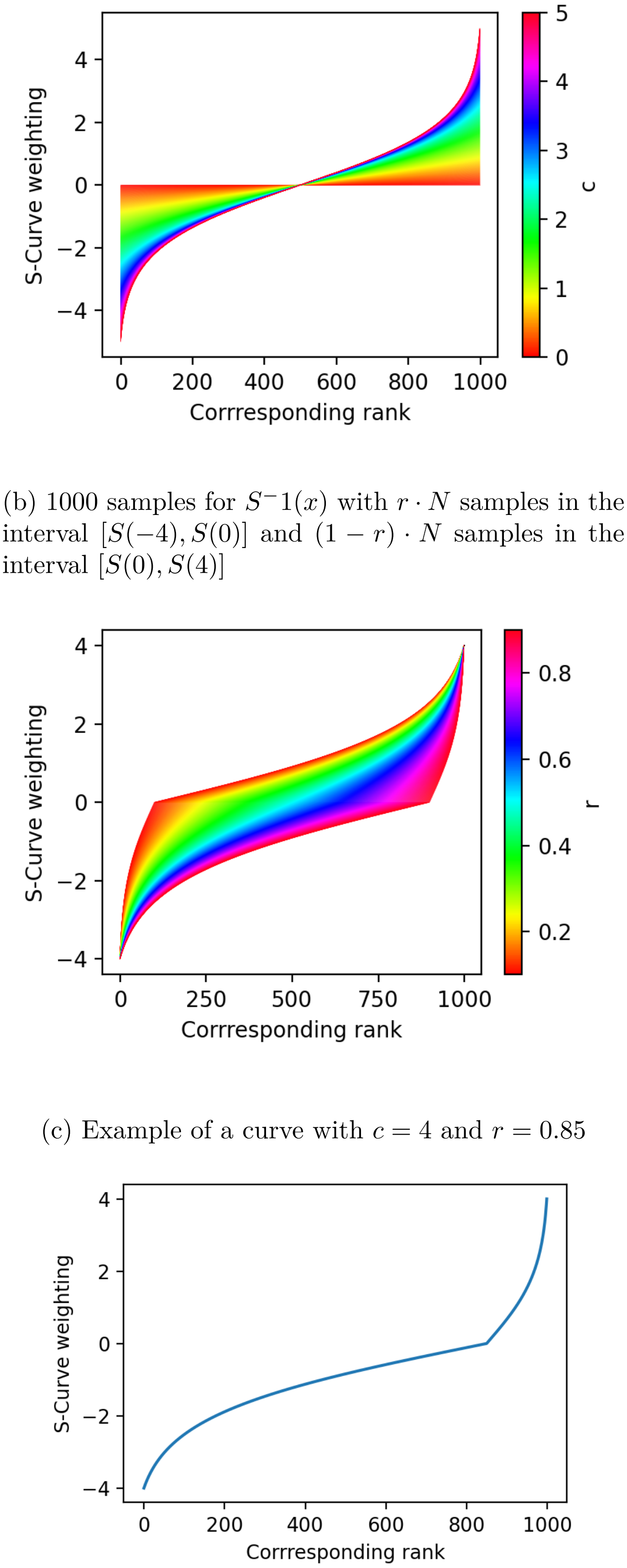
Figure 16: Distance values over rank, rank list created using the smoothed Manhattan distance, 500-MF tokens on the Brunet dataset



To break the curve symmetry and to be able to increase the conservation of the top rank while decreasing the conservation of the bottom ranked. The solution proposed is to have a new parameter r. $r \cdot N$ samples are taken for the left side of the curve $[S(-c), S(0)]$ And $(1 - r) \cdot N$ samples for the right side of the curve $[S(0), S(c)]$. Figure 17b shows the r parameter influence on a sigmoid with $c = 4$ and $r \in [0.1, 0.9]$. The r parameter must be in the interval $]0, 1[$. With a value close to 0 the soft-veto have a strong effect on the top ranked links. With a large value, a strong effect on the bottom ranked links, see Figure 17b.

To apply the soft-veto, each score in the rank list is multiplied by the curve at the corresponding rank. This soft-veto methodology aimed to be used on rank list before fusing, to have a more cleaved decision on the links positions. This strategy can be use on any type of scoring method (similarity, distances, Z-Score scores or probabilities).

Figure 17: Soft-veto curve parameters



(a) 1000 samples for $S^{-}1(x)$ in the interval $[S(-c), S(+c)]$

(b) 1000 samples for $S^{-}1(x)$ with $r \cdot N$ samples in the interval $[S(-4), S(0)]$ and $(1-r) \cdot N$ samples in the interval $[S(0), S(4)]$

(c) Example of a curve with $c = 4$ and $r = 0.85$

6.2 Evaluation

6.2.1 Rank Lists Fusion Evaluation

In this experiment, the goal is to understand if fusing individual rank lists can create better rank lists. The methodology used, is to create n isolated rank lists and fuse them for every r rank lists combinations (C_r^n possible rank lists combinations).

The rank list used are the ones obtained by using the nine retained rank list methods ($n = 9$), see Table 7 in Section 4.6. They represent the best configuration retained for each text representation. The rank list are fused using the two proposed methods: Z-Score fusion and regression fusion.

The r value is selected to 4, since it represents the number of different text representation. The number of possible fusion combinations with $n = 9$ and $r = 4$ is $C_4^9 = 126$. The 126 resulting fusions rank list are evaluated using the three following metrics: Average Precision (AP), R-Precision (RPrec) and High Precision (HPrec).

To be able to compare the rank list produced by fusion to the non-fused rank lists (one to many), two evaluation strategies are used. One called *Single-Max* which represent the best metrics for each individual rank list in the fusion and another one called *Single-Mean* which is represented by the mean of each rank list metrics in the fusion. More information on these two comparison strategy and an example are presented in Section 3.6.4.

If the rank lists produced by a fusion overcome the Single-Max evaluation, it means that the resulting rank list give results even better than every individual rank lists used for this fusion. This represents the best case where the fusion actually improve the results.

If the rank lists produced by a fusion overcome the Single-Mean evaluation, it means that the fused rank list have better results than the expected value of the individual rank lists (when a rank list is selected randomly). When using the fusion for unsupervised problems (such as clustering), it is a good approach since it gives stability.

For this experiment, the St-Jean A and St-Jean B corpora are used. The results of every fusion combinations using the proposed rank list and strategies are graphically presented in Figure 18.

Statistics on the 126 fusions on each metric and fusion schemes are resumed in Tables 15/16. These tables contain: the minimal value (min), the average and the standard deviation (Avg±Std), the maximal value, the argmin and argmax for the metrics (using ids of the rank list see Table 7 in Section 4.6). The reader can refer to Example 14 to clearly understand the values in these tables.

A sign test between the two strategies and the two fusion methods is presented in Table 17.

Using these metrics on St-Jean, the sign test tends to indicate that the fusion increase the overall quality of the rank list. The Z-Score fusion, produce for every metrics better results than both Single-Mean and Single-Max, the average precision is increased in average by $\sim$ 10% compared to the average of the average precision (Single-Mean) of the rank list used, and by around 6% in average when considering the maximal average precision (Single-Max).

In the other hand, the regression fusion, have an average increase of $\sim$ 3% in average precision for the Single-Mean and have a decrease of around 1% when comparing to the Single-Max. The standard deviation of the regression fusion is greater than the Z-Score fusion, this indicates a slightly greater instability in the results.

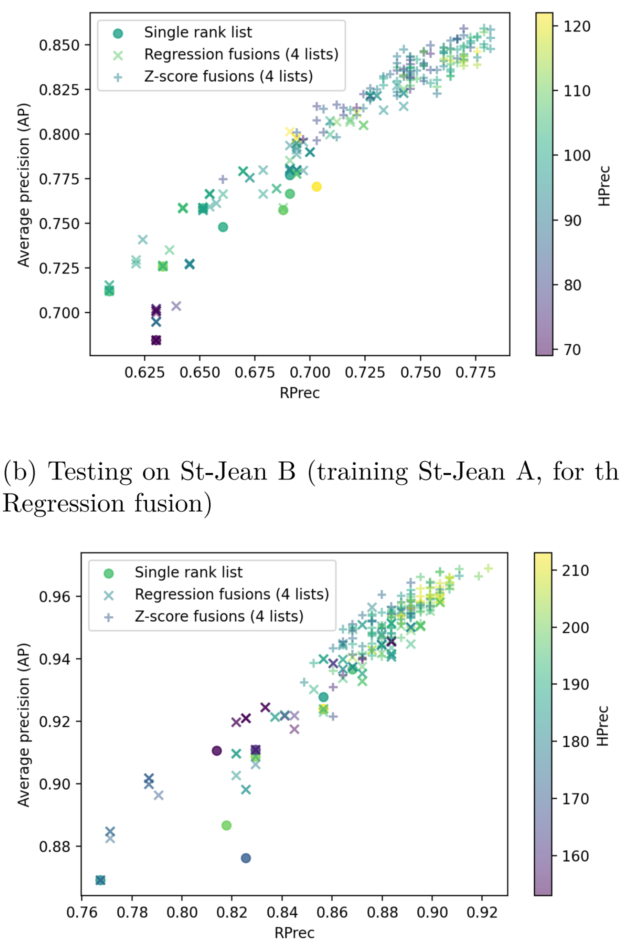
A general observation of the Tables 15 and 16 and Figure 18, show that the HPrec tends to be the least easy metric to increase when using the fusion. The results of fusing multiple rank lists when using regression fusion strategy tends to indicate that the resulting rank list has in average an equivalent quality as the best rank list used for the fusion, with some slight decreases. This can be further confirmed by looking at Table 17b, line Regression/T/Single-Max contains more ties than other lines.

No particular set of rank list tends to give the best results in every case when fusing with the two strategies. Every rank list is present in the argmax. With the Z-Score fusion, the best results are obtained with rank lists using different text representations. This observation will be further explorer in Section 6.2.4.

Overall, The Z-Score fusion give best results.

Figure 18: Evaluation of every combination of 4 rank lists fusions using Z-Score and Regression

(a) Testing on St-Jean A (training St-Jean B, for the Regression fusion)



(b) Testing on St-Jean B (training St-Jean A, for the Regression fusion)

6.2.2 Veto Evaluation

In this experiment, the goal is to understand if the veto method can improve the average precision on the fused rank lists when using the regression fusion method. To do so, the St-Jean retained rank list are used (ref. Table L in Annex). The fusion scheme is learnt on first St-Jean A and trained on St-Jean B (ref. Section 6.1.2), then inversely.

After computing the probabilities for each link in each rank list on the testing set, the probabilities under the threshold are set according to each proposed veto strategy (set to 0, -1, $-n$, $-\infty$).

An evaluation is done for the threshold ranging between 0.01 and 0.50, with a step of 0.01. A threshold of 0.50 is a theoretical upper bound. After a threshold of 0.50, the false link probability is smaller than the true link probability. Having a veto on a link more likely to be true link than

Example 14: Fusion evaluation methodology

Imagine a simple scenario where three individual rank lists are fused two by two. This corresponds to $C_2^3 = 3$ possible fusion combinations.

(a) Rank lists

The three individual rank lists have the following AP evaluation.

Rank lists	AP
Rank list 1	0.8
Rank list 2	0.9
Rank list 3	0.7

(b) Fusions

Now the rank lists are fused using a fusion scheme X. These create a single rank list for each rank list combinations. They can be evaluated with the average precision. For the example, they obtain the score written in column *Fusion X AP* of this following table.
We can also compute the rank list aggregations using the Single-Mean / Single-Max schemes for each rank list combinations.

Rank lists fusions	Fusion X AP	Single-Mean AP	Single-Max AP
Rank list 1 and 2	0.7	$(0.8 + 0.9)/2 = 0.85$	$max(0.8, 0.9) = 0.9$
Rank list 2 and 3	0.8	$(0.9 + 0.7)/2 = 0.80$	$max(0.9, 0.7) = 0.9$
Rank list 1 and 3	0.9	$(0.8 + 0.7)/2 = 0.75$	$max(0.8, 0.7) = 0.8$

(c) Statistics

Then we can compute statistics for all the rank lists combinations (each line from the previous table).

Statistics	Fusion X AP	Single-Mean AP	Single-Max AP
Min	0.7	0.75	0.8
Mean±Std	0.8±0.1	0.8±0.05	0.87±0.06
Max	0.9	0.85	0.9
Argmin	[1, 2]	[1, 3]	[1, 3]
Argmax	[1, 3]	[1, 2]	[1, 2]

Instead of only evaluating the average precision, multiple metrics are used for the real experiment (this creates additional columns). The values are also presented differently in the real experiment since two fusion strategies are used.

(d) Sign Test

Each rank lists combination (each line from the fusion table) can be used to create a sign test. If the rank list by fusion is better for one combination than Single-Mean, the fusion earn 1 point. If the rank list by fusion have the same results as Single-Mean, it is a tie. Otherwise, the Single-Mean earn 1 point. The same goes for Single-Max.
Here is the sign test for the example.

Sign Test	AP
Fusion X/Tie/Single-Mean	1/1/1
Fusion X/Tie/Single-Max	1/0/2

false link is a bad idea.

Rank lists regression fusion are normally computed with the probabilities average, with one small difference for the $-\infty$ strategy when a $-\infty$ is encountered in the mean, the mean will always

Table 15: Fusion statistics on St-Jean A

(a) Single-Mean

Stats	AP	RPrec	HPrec
Min	0.72	0.63	56.25
Avg±Std	0.74 ± 0.01	0.66 ± 0.01	70.78 ± 8.00
Max	0.77	0.69	84.25
Argmin	[1,4,6,7]	[4,5,6,7]	[0,1,5,7]
Argmax	[0,2,5,8]	[0,2,3,8]	[2,3,4,6]

(b) Single-Max

Stats	AP	RPrec	HPrec
Min	0.75	0.65	72.00
Avg±Std	0.77 ± 0.01	0.69 ± 0.01	88.66 ± 9.55
Max	0.78	0.70	99.00
Argmin	[1,4,6,7]	[4,5,6,7]	[0,1,5,7]
Argmax	[0,1,2,3]	[0,1,2,3]	[0,1,2,3]

(c) Z-Score fusion

Stats	AP	RPrec	HPrec
Min	0.77	0.66	69.00
Avg±Std	0.84 ± 0.02	0.75 ± 0.02	93.92 ± 11.15
Max	0.86	0.78	122.00
Argmin	[0,4,5,7]	[0,4,5,7]	[0,1,4,5]
Argmax	[0,3,6,7]	[0,3,4,8]	[0,3,7,8]

(d) Regression fusion (training St-Jean B)

Stats	AP	RPrec	HPrec
Min	0.68	0.61	20.00
Avg±Std	0.76 ± 0.04	0.67 ± 0.04	68.84 ± 19.41
Max	0.83	0.74	112.00
Argmin	[2,3,7,8]	[1,2,4,8]	[2,3,4,7]
Argmax	[0,1,2,6]	[0,1,2,3]	[0,1,3,7]

Table 16: Fusion statistics on St-Jean B

(a) Single-Mean

Stats	AP	RPrec	HPrec
Min	0.89	0.81	137.50
Avg±Std	0.91 ± 0.01	0.83 ± 0.01	154.89 ± 7.22
Max	0.92	0.85	172.00
Argmin	[3,4,6,8]	[1,3,6,8]	[1,6,7,8]
Argmax	[0,2,5,7]	[0,2,5,7]	[0,3,4,5]

(b) Single-Max

Stats	AP	RPrec	HPrec
Min	0.91	0.83	153.00
Avg±Std	0.93 ± 0.01	0.86 ± 0.01	174.75 ± 7.29
Max	0.94	0.87	182.00
Argmin	[3,4,6,8]	[1,3,6,8]	[1,6,7,8]
Argmax	[0,1,2,3]	[0,1,2,3]	[0,1,2,5]

(c) Z-Score fusion

Stats	AP	RPrec	HPrec
Min	0.92	0.85	153.00
Avg±Std	0.95 ± 0.01	0.89 ± 0.01	188.91 ± 12.87
Max	0.97	0.92	213.00
Argmin	[2,3,6,8]	[3,4,6,8]	[1,2,3,6]
Argmax	[1,2,5,7]	[1,2,5,7]	[0,3,4,7]

(d) Regression fusion (training St-Jean A)

Stats	AP	RPrec	HPrec
Min	0.87	0.77	124.00
Avg±Std	0.93 ± 0.02	0.86 ± 0.04	165.25 ± 22.09
Max	0.97	0.91	210.00
Argmin	[2,3,6,8]	[2,3,6,8]	[1,2,3,8]
Argmax	[0,1,2,7]	[0,1,2,7]	[0,1,3,7]

be $-\infty$. To understand if the veto improve the results, the average precision gain is compute between the rank list with and without the veto strategy.

The average precision gain over the threshold value is visually presented in Figure 19. The best configurations found are summarized in Table 18.

Overall, the veto strategy seem to be a bad option. The best result obtained have only a 0.0193 gain in average precision, with a threshold of 0.01, which is the closest to the non threshold approach, since only a small portion of the rank list is affected. This tiny improvement may be due to random factors. Using the $-\infty$ strategy only give worse results.

The veto strategy is only impactful when most of the rank lists rank a false link at the top, except one rank list which correctly identify this false link and rank him at the bottom. The rank list used for the experiment do not have such special case, since most of them correct classify false link at the bottom and true links at the top. This can be one possible reason on why the veto fail to improve the results. This strategy may be appliable on harder corpus such as the PAN 16 corpus or with even more diverse rank list generation strategies.

6.2.3 Soft-veto Evaluation

To evaluate the soft veto strategy performances, the same strategy as for the veto strategy is used. The average precision gain obtained when using the soft-veto over the fusion without veto is com-

Table 17: Rank lists fusion sign test. *The star (*) indicate a Binomial test p-value smaller than 5%*

(a) Testing St-Jean A (training St-Jean B, for the Regression fusion)

Metric	AP	RPrec	HPrec
Z-Score/Tie/Single-Mean	*126/0/0	*126/0/0	*124/0/2
Z-Score/Tie/Single-Max	*125/0/1	*124/1/1	*77/4/45
Regression/Tie/Single-Mean	*85/0/41	66/1/59	58/1/67
Regression/Tie/Single-Max	58/0/68	38/7/81*	16/5/105*

(b) Testing St-Jean B (training St-Jean A, for the Regression fusion)

Metric	AP	RPrec	HPrec
Z-Score/Tie/S-Mean	*126/0/0	*126/0/0	*126/0/0
Z-Score/Tie/S-Max	*125/0/1	*123/2/1	*108/2/16
Regression/Tie/S-Mean	*116/0/10	*103/1/22	*87/1/38
Regression/Tie/S-Max	*87/0/39	*72/11/43	43/9/74*

Figure 19: Veto strategies evaluations on St-Jean A and B depending on the threshold

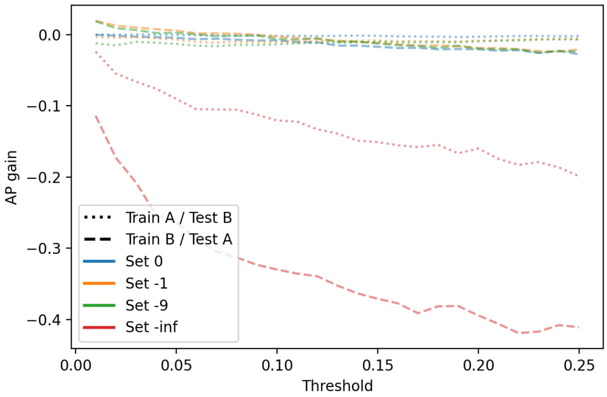


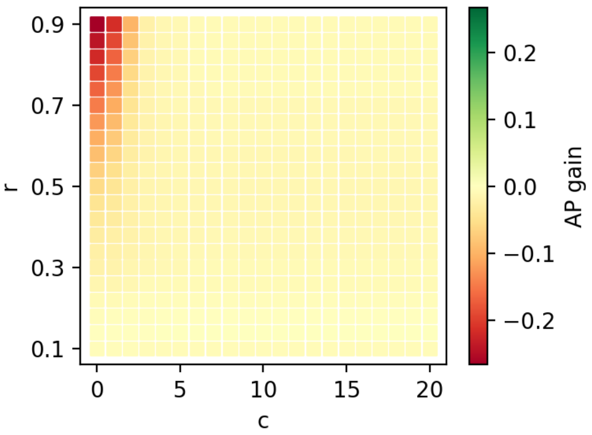
Table 18: Maximal gain on the average precision using the veto and Argmax

	Train,Test	
	Train A, Test B	Train B, Test A
Set to 0	1.98e-04/0.03	-9.11e-05/0.01
Set to -1	-3.28e-03/0.01	1.93e-02/0.01
Set to $-n$	-6.73e-03/0.24	1.88e-02/0.01
Set to $-\inf$	-2.40e-02/0.01	-1.14e-01/0.01

puted. The Z-Score fusion strategy is used for this experiment.

The soft-veto has two parameters: c and r, to be able to find the best parameters the grid search approach is used. Since the c parameter must be in the interval $]0, \infty[$ and the r parameter must be in the interval $]0, \infty[$, the grid is constructed from the Cartesian product of the two following sets. $c = \{0 + \epsilon, 1, 2, ..., 19, 20\}$, with ϵ a small value (for example 10^{-6}) and $r = \{0.10, 0.14, ..., 0.86, 0.90\}$. When computing the soft-veto gain using the grid search on the retained rank lists from Oxquarry (see Table 7), Figure 20 is obtained.

Figure 20: Average precision gain with soft veto s-curve on Oxquarry



The best parameters are $c = 18$ and $r = 0.14$ with an average precision gain of $3.50e - 04$. Like for the veto experiment, no real average precision gain can be obtained using the soft-veto strategy. The average precision gain is small and can be due to random factors.

6.2.4 Average Precision Fusion Gain Relation with the Rank Lists Diversity

As stated previously, we believe fusing high quality rank lists (high AP), with the most diverse results can increase the quality of the rank list obatined by fusion. In this section, we aim to provide even more clues to motivate this assumption. First, a correlation metric to compare the rank lists must be selected. Secondly, we compare the correlation coefficients to the average precision gain to verify the assumption.

To compute a correlation between two rank lists, the Kendall rank correlation coefficient is used [33]. The Kendall rank correlation computes a correlation index based on the ordinal association between the two rank lists by computing the number of swaps needed to obtain one rank list from the other. The fewer swaps needed, the closer the rank lists are.

The default weighing strategy called *hyperbolic weighing* is used for the Kendall rank correlation coefficient. It favours top ranks in the same fashion as the reciprocal rank $(1/(r+1))$. Swaping top ranks are more important for the Kendall coefficient than bottom ranks.

The Kendall coefficient is ranged between -1 and 1, with 1 being an identical correlation and -1 a completely different correlation. If our assumption is correct, the average precision gain obtained by fusing two rank lists should be negatively correlated to the rank list Kendall correlation coefficient. A large gain, when the rank lists are different, thus a low rank list correlation coefficient, and vice versa.

The fusion strategy used here is the Z-Score fusion, since this strategy is parameterless and obtained satisfying results during previous experiments. As for previous experiments dealing with fusion, the gain in average precision when comparing a set of rank list to a fused rank list can be established in the two following ways: *Single-Mean* and *Single-Max*. Single-Mean represent the rank list mean average precision and Single-Max the rank list maximal average precision. Example 4 in Section 3.6.4 contain a small example with two rank lists for the Single-Mean and Single-Max methods to compute the average precision gain.

Table 19 show the Kendall correlation coefficient for every pairwise rank lists combination in St-Jean (retained rank lists, see Table 7 in Section 4.6).

The most similar rank lists pair are the two rank list using letters n-grams (0.95, id 4 and 5). This result can be easily understood, since both rank list use a similar text representation and distance measure. They thus obtain similar rank lists.

The least similar rank lists pair are the BZip2 compression and the 250-MF 2-POS (0.66, id 6 and 7). The compression method is the simplest approach, this method does not require to understand the text to create the rank list. As for the 250-MF 2-POS rank list, to create the rank list, it requires two level of understanding. First the parser need to understand texts to create the POS, and the second level are the short sequences which involve an even deep understanding of the text. Due to this large difference in rank list construction, the two rank lists obtain a small correlation coefficient.

For every pair of retained rank lists in St-Jean, the Single-Mean fusion gain is computed and is compared to the correlation coefficient in Figure 21. Using the linear regression, the best line for these point is computed [33]. After reproducing this experiment for every corpus and fusion gain computation methods, the results in Table 20 are obtained.

For all experiment the linear correlation coefficient, r-value, is negative, which indicate that the average precision gain is negatively correlated to the Kendall correlation coefficient between the rank lists. Thus, the assumption that suggested that the more diverse the rank lists are, the higher the quality of the fusion, seem to be confirmed. The p-value for the linear regression (null hypothesis: the slope is 0) obtained after the regression is below 5% for every of the Single-Mean experiments and for the Single-Max St-Jean corpus. A clear linear relationship between the average precision gain and rank list diversity is observed.

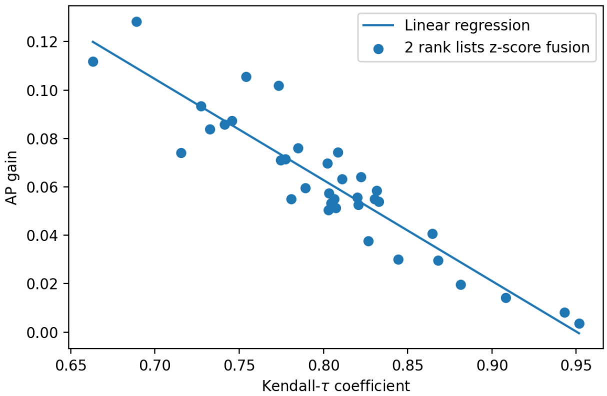
Using the Single-Max strategy on Oxquarry and Brunet give a p-value of 15.1% and 7.04% respectively. This can indicate that the linear correlation between the average precision gain and the diversity of the rank list can be due to random factors.

Since the Single-Mean average precision gain follow a linear model, the gain can be predicted using

Table 19: Pairwise Kendall correlation coefficient on St-Jean retained rank lists

id	1	2	3	4	5	6	7	8
0	0.86	0.83	0.80	0.88	0.91	0.73	0.81	0.75
1	-	0.87	0.83	0.80	0.83	0.79	0.73	0.77
2	-	-	0.84	0.78	0.80	0.82	0.72	0.79
3	-	-	-	0.78	0.81	0.80	0.75	0.85
4	-	-	-	-	0.95	0.77	0.81	0.75
5	-	-	-	-	-	0.78	0.82	0.77
6	-	-	-	-	-	-	0.66	0.79
7	-	-	-	-	-	-	-	0.82

Figure 21: Fusion average precision gain (using Single-Mean scheme) over rank list distance on St-Jean



the slope and the y-intercept of the linear regression. In an unsupervised model, computing the Kendall correlation coefficient can allow to optimize the fusion gain. The lowest the correlation coefficient, the greater the average precision gain should be.

This approach have one main draw back. The rank list used have to be known has *good rank list*. Otherwise, if a *bad rank list* and a *good rank list* are compared using Kendall correlation coefficient, they will obtain a low correlation, since *bad answers* are often very different from *good answers*. This low correlation coefficient should indicate a positive good gain. But when fused, the gain will not be positive, since fusing *bad rank lists* and *good rank lists* tends to produce terrible results.

6.3 Fusion summary

The Z-Score fusion and regression fusion was shown in this study to improve rank lists quality when fusing *good* rank list together.

Table 20: Linear regression on the average precision gain over Kendall r-coefficient

(a) Single-Mean

Corpus/Method	r-value	p-value	Std. error
Oxquarry	-0.63	2.04e-03	4.36e-02
Brunet	-0.89	7.14e-08	2.17e-02
St-Jean	-0.93	9.30e-17	2.49e-02

(b) Single-Max

Corpus/Method	r-value	p-value	Std. error
Oxquarry	-0.32	1.51e-01	8.15e-02
Brunet	-0.40	7.04e-02	4.50e-02
St-Jean	-0.86	1.60e-11	3.72e-02

The rank lists obtained by fusing every retained rank lists with the Z-Score method are referred as *Z-Score rank lists*. This corresponds to one rank list for each corpus.

When fusing rank lists with the regression fusion, pairs of corpora are required (one for training and one for testing). When evaluating this method with four corpora (Oxquarry, Brunet, St-Jean A and St-Jean B), this creates $4^2 = 16$ possible corpus pairs which yield one rank list each. These two methods are summarized in the schema in Figure 22.

Table O in Annex contain the rank list evaluation, for every corpus, with every retained rank lists for the fusion. The average precision obtained with the Z-Score rank lists is greater than each individual methods for every corpus except for the Oxquarry corpus. The Oxquarry Z-Score rank list have an average precision of 0.84 but using the individual methods: *the Cosine distance or the Clark distance with the* 750*-MF tokens* gives an average precision of 0.89. The corpus with the worst average precision using the Z-Score rank lists is obtained on Brunet with 0.76 and is as efficient as the individual *BZip2 compression with CBC distance* method. Even though here the Z-Score rank list have lower quality than an individual method, as stated previous, in an unsupervised environment this can not be detected (average precision require labels). For unsupervised tasks, we advise to still use the Z-Score fusion, since we showed that it provides in average better results than each individual methods.

Table 21 is a summary of the results for the fusion using every individual methods. For both methods, the mean average precision across all cor-

Table 21: Fusion evaluation summary, mean across every corpora

	AP	RPrec	HPrec
Z-Score fusion	0.85	0.78	101.50
Regression fusion	0.85	0.78	98.50
Individual methods average	*0.79*	*0.71*	*78.88*

pora reach 0.85. The mean average precision for the fused rank lists is 7% better using the fusion than the individual methods average ($AP = 0.79$). Since the regression fusion require a training corpus and had shown some weakness during its evaluation. We recommend using the Z-Score fusion instead.

The proposed veto methods did not provide good results and are not advised to use.

6.4 Clustering with Fusion Evaluation

Table P / Q / R / S in Annex show the evaluation of the clustering task for rank lists obtained by fusing every retained rank lists using the Z-Score method.

Table 22 is a summary of the results. The upper bound is, using the same rank lists, the best clustering achievable with the hierarchical clustering for specific linkage criterion. It is obtained by evaluating the hierarchical clustering at each step of the algorithm. The clustering result with the greatest metrics is the upper bound for the hierarchical clustering.

With the Z-Score fusion the upper bound rise from 0.88 to 0.92 with the average linkage criterion, this corresponds to a $\sim$ 5% relative increase. For the complete linkage, the $B^3_{F_1}$ rise from 0.87 to 0.89, which correspond to a $\sim$ 2% relative increase. With the Z-Score fusion rank lists, every clustering models have its results improved over the individual approaches. This result further motivate the rank list fusion usefulness.

The Silhouette-based clustering method with the average linkage criterion yield the best results across the four models. This result is only slightly better than the distribution-based model with the complete linkage criterion. This model was the one that obtained the best results for the clustering on the individual rank lists methods.

Table 22: Retained rank lists Mean $B^3_{F_1}/r_{diff}$ for each corpus pair

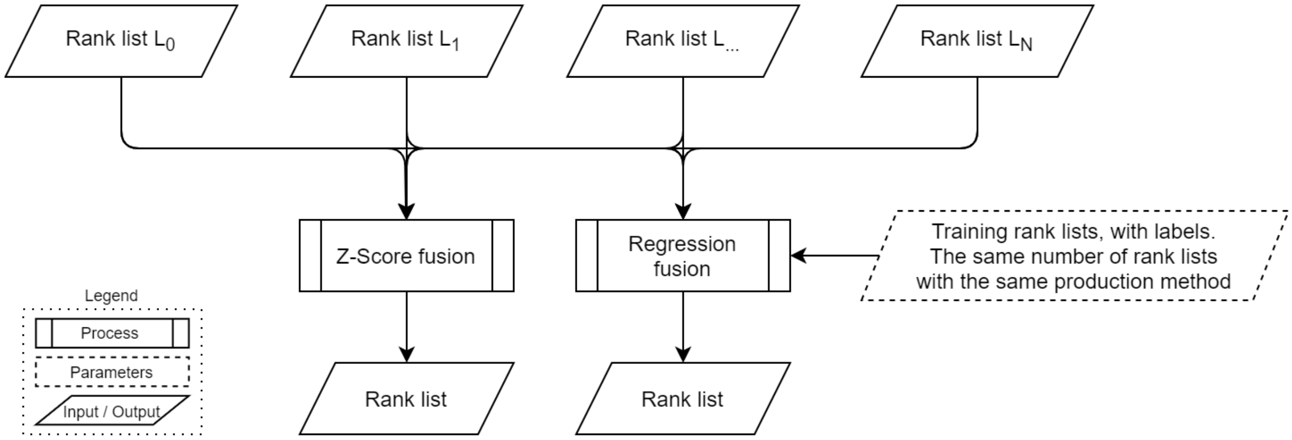
	Linkage criterion	
Clustering method	Average	Complete
Silhouette-based ($\alpha = 0$)	**0.75/0.16**	**0.75/0.16**
Silhouette-based ($\alpha = -0.2$)	**0.87/0.07**	0.84/0.08
Distribution-based	0.83/-0.02	**0.86/0.04**
Regression-based	**0.84/0.07**	0.80/0.13
Rank lists upper bound	*0.92/0.00*	*0.89/0.00*

For the individual methods, the average linkage criterion is the best criterion for each method, except the distribution-based clustering, which obtain better results with the complete linkage.

The $B^3_{F_1}$ achieved for each model with the best linkage criterion is 5 to 10% worse than the upper bound, if we exclude the Silhouette-based model (with $\alpha = 0$) which is 18% worse than the upper bound. Excluding the non-tweak Silhouette-based model clustering, every other proposed clustering model seem to give reasonable results with fused rank lists.

The clustering with the fused rank list is in average 6% better than the clustering with the individual methods.

Figure 22: Fusion methods schema



Chapter 7

Conclusion and Future Work

Evaluation is an important aspect to improve our knowledge. In this view, this study is based on three known collections namely Oxquarry, Brunet and St-Jean. These three corpora containing excepts of literature novels. Nine different techniques to produce individual rank lists are cherry-picked for the clustering and fusion steps. These techniques are mostly based on the most frequent words. Compression techniques are also evaluated and used.

Once each intermediate representations of the corpora, namely the rank lists, are created and evaluated. A hierarchical clustering model is applied. With this model, a cut in the rank list which discriminate the same authors documents to different authors documents is required. Three solutions to find the best cut position are proposed.

The first is a fully unsupervised method based on the Silhouette score. We added an extra hyper-parameter to the Silhouette-based clustering and tweaked it for the authorship clustering problem. By doing so, this model improves its clustering results in average by 7% across the three corpora.

The second is a supervised method and thus require a corpus with labels to calibrate the system. We called this method distribution-based. This method output a single numerical value which is used to perform the cut for any subsequent corpus. This strategy gives the best results for the individual methods.

The last technique to perform the cut is also supervised and is based on a logistic regression model. For any new corpus, this model is used to predict the best position in the rank list for the cut.

The three methods to find the best cut for the hierarchical clustering give good results close to the best clustering achievable for the rank lists used. Additionally, we demonstrated a correlation between the rank lists quality and the clustering quality.

In a second part of this study, two rank list fusion method are proposed and evaluated. This proposition come from the idea that combining good rank lists can give one better than each individual one. One fusion technique uses the Z-Score normalization and the other one convert the rank list scores into probabilities using multiple logistic regression. The Z-Score fusion technique show a significant improvement of the results when compared to the best rank list used for the fusion. Whereas the logistic regression fusion procedure give slightly worse results but still shown a significant stabilization of the results. The rank list produced with the logistic regression fusion give a rank list quality equivalent to the average quality of the rank list used. These results motivate the rank list fusion's beneficial aspect for unsupervised task such as the authorship clustering problem.

Two veto-based techniques are proposed and evaluated to try to further improve the fused rank list's quality. No real improvements are found using the proposed veto strategies on high quality rank lists (more than 0.7 in AP).

To further motivate the strength of the rank list fusion, we showed a correlation between the rank lists variety and the improvements provided when fusing rank lists. We showed that by using for the fusion, rank lists produced along different text representations increase more the effectiveness of resulting rank list than using rank lists with the same text representation but with different distance measures.

At the end of this study, we evaluated the clustering models using the rank list obtained by fusion. As we expected from previous results, the clustering obtained with the rank lists obtained by

Z-Score fusion is better than the individual methods. We obtained overall a 6% improvement using the Z-Score fusion over the individual methods. This improvement in the clustering is due to the fact that the rank lists obtained by fusion are better than the ones obtained by individual methods and thus, as we showed previously, having good rank lists tends to produce good clusterings. The tweaked Silhouette-based clustering unsupervised method gives the best results for the rank lists obtained by Z-Score fusion.

The authorship clustering models used still need to be compared to other models from the state of the art. This can be done by testing this model with corpora written in other languages or extracted from domains or support (e.g. blogs), which were also used for this task.

As the literature shows, many methods to generate rank lists are possible. We showed that using the most diverse individual technique, improve the fusion quality. Thus, testing our current fusion schemes with other individual methods, could potentially further improve the fusion results.

The N-First characters text representation gave outstanding results on some corpora. Though it was not retained for the fusion, since for other corpora it gave poor results. A deeper analysis has to be made for this text representation.

Another point not treated in this study is to use compression techniques with different text representation. The POS sequences seem a good choice to explore with compression techniques.

Other fusion methods still need to be tested to potentially further increase fusions quality. For example: other normalization techniques can be used and/or other ways to compute a central tendency (e.g. the geometric mean or the harmonic mean).

Veto-based technique were experimented to force bottom ranked links according to one rank list to be at the bottom of a fused rank list. An alternative to this veto, could be to force top rank lists, to be at the top of the fused rank list. This technique still have to be explored and may have more impactful results.

Bibliography

[1] **Anne Le Calvé and Jacques Savoy (2000)**, *Database Merging Strategy based on Logistic Regression*, Information Processing & Management, Volume 36, Issue 3, 1 May 2000, Pages 341-359.

[2] **Vlado Keselj, Fuchun Peng, Nick Cercone, Calvin Thomas (2003)**, *N-Gram-based Author Profiles for Authorship Attribution*, 2003 Pacific Association for Computational Linguistics.

[3] **Enrique Amigo Julio Gonzalo, Javier Artiles, Felisa Verdejo (2009)**, *A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints*, Departamento de Lenguajes y Sistemas Informaticos, UNED, Madrid, Spain, 2009.

[4] **Robert Layton, Paul Watters and Richard Dazeley (2011)**, *Automated Unsupervised Authorship Analysis using Evidence Accumulation Clustering*, 2011 Natural Language Engineering 19.

[5] **Shlomo Argamon and Patrick Juola (2011)**, *Overview of the International Authorship Identification Competition at PAN-2011*, Authorship Verification task at PAN @ CLEF 2011.

[6] **W. Oliveira Jr., E. Justino, L.S. Oliveira (2013)**, *Comparing Compression Models for Authorship Attribution*, Forensic Science International 228 (2013) 100-104.

[7] **Jacques Savoy (2015)**, *Estimating the Probability of an Authorship Attribution*, Journal of the association for information science and technology, 2015.

[8] **Jacques Savoy (2015)**, *Text Representation Strategies: An Example With the State of the Union Addresses*, Journal of the association for information science and technology, 2015.

[9] **Mirco Kocher and Jacques Savoy (2016)**, *A Simple and Efficient Algorithm for Authorship Verification*, Journal of the association for information science and technology, 2016.

[10] **Efstathios Stamatatos, Michael Tschuggnall, Ben Verhoeven, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast (2016)**, *Clustering by Authorship Within and Across Documents*, Author Clustering task at PAN @ CLEF 2016.

[11] **Douglas Bagnall (2016)**, *Authorship Clustering using Multi-headed Recurrent Neural Networks*, Author Clustering task at PAN @ CLEF 2016.

[12] **Mirco Kocher (2016)**, *UniNE at CLEF 2016: Author Clustering*, Author Clustering task at PAN @ CLEF 2016.

[13] **Jawwad A. Shamsi, Sherali Zeadally, Fareha Sheikh and Angelyn Flowers (2016)**, *Attribution in cyberspace: techniques and legal implications*, Security and communication networks, 2016

[14] **Dominique Labbé (2017)**, *Une Expérience d'Attribution d'Auteur. Le Corpus Saint-Jean*, 2017 CNRS – Université de Grenoble-Alpes.

[15] **Dan Salvato, Team Salvato (2017)**, *Doki Doki Literature Club!*, Freeware, visual novel.

[16] **Mirco Kocher and Jacques Savoy (2018)**, *Evaluation of Text Representation Schemes and Distance Measures for Authorship Linking*, Digital Scholarship in the Humanities, The Author(s), 2018.

[17] **Jacques Savoy (2018)**, *Is Starnone really the author behind Ferrante?*, Digital Scholarship in the Humanities, Vol. 33, No. 4, 2018.

[18] **OECD - Social Policy Division - Directorate of Employment, Labour and Social Affairs (2019)**, *SF2.3: Age of Mothers at Child Birth and Age-specific Fertility*, 2019 OECD Family Database.

[19] **Emir Araujo-Pino, Helena Gómez-Adorno, and Gibran Fuentes-Pineda (2020)**, *Siamese Network applied to Authorship Verification*, Authorship verification task at PAN @ CLEF 2020.

[20] **Janith Weerasinghe and Rachel Greenstadt (2020)**, *Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification*, Authorship verification task at PAN @ CLEF 2020.

[21] **Catherine Ikae (2020)**, *UniNE at PAN-CLEF 2020: Author Verification*, Authorship verification task at PAN @ CLEF 2020.

[22] **Catherine Ikae, Jacques Savoy (2020)**, *UniNE at PAN-CLEF 2020: Profiling Fake News Spreaders on Twitter*, Profiling Fake News Spreaders on Twitter task at PAN @ CLEF 2020.

[23] **Mike Kestemont, Enrique Manjavacas, Ilia Markov, Janek Bevendorff, Matti Wiegmann, Efstathios Stamatatos, Martin Potthast, and Benno Stein (2020)**, *Overview of the Cross-Domain Authorship Verification Task at PAN 2020*, Authorship verification task at PAN @ CLEF 2020.

[24] **Jacques Savoy (2020)**, *Machine Learning Methods for Stylometry: Authorship Attribution and Author Profiling*, Springer Nature Switzerland AG 2020. ISBN: 978-3-030-53359-5.

[25] **Webis Group**, *PAN@CLEF*: `https://pan.webis.de/` (last access: 27.05.2021), Pan.webis.de.

[26] **Stamatatos Efstathios, Tschuggnall Michael, Verhoeven Ben, Daelemans Walter, Specht Günther, Stein Benno, Potthast Martin**, *PAN16 Author Identification: Clustering*: `https://zenodo.org/record/3737587` (last access: 30.03.2021), Zenodo.org.

[27] **Stamatatos Efstathios, Juola Patrick, Potthast Martin, Stein Benno, Verhoeven Ben**, *PAN16 Author Clustering 2016*: `https://pan.webis.de/clef16/pan16-web/author-clustering.html` (last access: 07.04.2021), Webis.de.

[28] **University of Neuchâtel**, *UNINE Corpus et sites internets*: `http://www.unine.ch/clc/home/Corpus.html` (last access: 30.03.2021), Unine.ch.

[29] **Python**, *Python is a programming language that lets you work quickly and integrate systems more effectively.*: `https://www.python.org/` (last access: 07.04.2021), python.org.

[30] **Python Standard Library**, *Data Compression and Archiving*: `https://docs.python.org/3/library/` (last access: 20.04.2021), python.org.

[31] **Matplotlib**, *Visualization with Python*: `https://matplotlib.org/` (last access: 07.04.2021), matplotlib.org.

[32] **Numpy**, *The fundamental package for scientific computing with Python*: `https://numpy.org/` (last access: 07.04.2021), numpy.org.

[33] **SciPy**, *SciPy a Python-based ecosystem for mathematics, science, and engineering*: `https://www.scipy.org/` (last access: 07.04.2021), scipy.org.

[34] **Scikit-learn**, *Machine Learning in Python*: https://scikit-learn.org/ (last access: 07.04.2021), scikit-learn.org.

[35] **Wikipédia (en)**, *Silhouette (clustering)*: https://en.wikipedia.org/wiki/Silhouette_(clustering) (last access: 16.06.2021), wikipedia.org.

[36] **Wikipédia (en)**, *Evaluation measures (information retrieval)*: https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval) (last access: 05.07.2021), wikipedia.org.

[37] **Wikipédia (en)**, *F-score*: https://en.wikipedia.org/wiki/F-score (last access: 14.07.2021), wikipedia.org.

[38] **Project Jupyter**, *To develop open-source software, open-standards, and services for interactive computing across dozens of programming languages*: https://jupyter.org/ (last access: 21.05.2021), jupyter.org.

[39] **Hugo Hromic (hhromic)**, *Simple Extended BCubed implementation in Python for clustering evaluation*: https://github.com/hhromic/python-bcubed (last access: 06.05.2021), GitHub Page.

[40] **Ilya Flyamer (Phlya)**, *A small library for automatically adjustment of text position in matplotlib plots to minimize overlaps*: https://github.com/Phlya/adjustText (last access: 06.05.2021), GitHub Page.

[41] **Casper da Costa-Luis (casperdcl) and Stephen Karl Larroque (lrq3000)**, *A Fast, Extensible Progress Bar for Python and CLI*: https://github.com/tqdm/tqdm (last access: 17.05.2021), GitHub Page.

Appendices

Definitions and Corpora

Table A: General information and statistics on the PAN @ CLEF corpus

(a) Training set

Name	Lang.	Authors	Texts	r	True Links	Links	tl_r	Avg. #Tokens	Avg. Token size
01	en	35	50	0.7	26	1225	0.021	872.0	4.161
02	en	25	50	0.5	75	1225	0.061	881.0	4.118
03	en	43	50	0.86	8	1225	0.007	867.0	4.113
04	en	55	80	0.688	36	3160	0.011	1125.0	4.283
05	en	70	80	0.875	12	3160	0.004	1252.0	4.318
06	en	40	80	0.5	65	3160	0.021	1180.0	4.305
07	nl	51	57	0.895	7	1596	0.004	1261.0	4.582
08	nl	28	57	0.491	76	1596	0.048	1533.0	4.648
09	nl	40	57	0.702	30	1596	0.019	1184.0	4.613
10	nl	54	100	0.54	77	4950	0.016	145.0	4.41
11	nl	67	100	0.67	46	4950	0.009	152.0	4.393
12	nl	91	100	0.91	10	4950	0.002	142.0	4.386
13	gr	28	55	0.509	38	1485	0.026	903.0	4.596
14	gr	38	55	0.691	25	1485	0.017	895.0	4.633
15	gr	48	55	0.873	8	1485	0.005	879.0	4.62
16	gr	50	55	0.909	6	1485	0.004	653.0	4.318
17	gr	28	55	0.509	55	1485	0.037	781.0	4.34
18	gr	40	55	0.727	19	1485	0.013	707.0	4.298

(b) Testing set

Name	Lang.	Authors	Texts	r	True Links	Links	tl_r	Avg. #Tokens	Avg. Token size
01	en	50	70	0.714	33	2415	0.014	666.0	4.327
02	en	35	70	0.5	113	2415	0.047	678.0	4.241
03	en	64	70	0.914	7	2415	0.003	663.0	4.324
04	en	58	80	0.725	30	3160	0.009	1168.0	4.284
05	en	72	80	0.9	10	3160	0.003	1186.0	4.301
06	en	42	80	0.525	68	3160	0.022	1156.0	4.287
07	nl	42	57	0.737	24	1596	0.015	1365.0	4.63
08	nl	50	57	0.877	8	1596	0.005	1377.0	4.576
09	nl	30	57	0.526	65	1596	0.041	1110.0	4.561
10	nl	88	100	0.88	16	4950	0.003	171.0	4.5
11	nl	51	100	0.51	76	4950	0.015	169.0	4.448
12	nl	71	100	0.71	37	4950	0.007	175.0	4.444
13	gr	50	70	0.714	24	2415	0.01	864.0	4.624
14	gr	35	70	0.5	52	2415	0.022	898.0	4.603
15	gr	62	70	0.886	9	2415	0.004	887.0	4.603
16	gr	51	70	0.729	24	2415	0.01	550.0	4.236
17	gr	64	70	0.914	7	2415	0.003	542.0	4.229
18	gr	37	70	0.529	44	2415	0.018	672.0	4.262

Table B: Excerpts from the Oxquarry corpus

Id	**Author**	**Short title**
A1	Hardy	Jude
A2	Butler	Erewhon
B1	Butler	Erewhon
B2	Morris	Dream of JB
C1	Morris	News
C2	Tressel	Ragged TP
D1	Stevenson	Catriona
D2	Hardy	Jude
E1	Butler	Erewhon
E2	Stevenson	Ballantrae
F1	Stevenson	Ballantrae
F2	Hardy	Wessex Tales
G1	Conrad	Lord Jim
G2	Orczy	Elusive P
H1	Hardy	Madding
H2	Conrad	Lord Jim
I1	Orczy	Scarlet P
I2	Morris	News
J1	Morris	Dream of JB
J2	Hardy	Well-beloved
K1	Stevenson	Catriona
K2	Conrad	Almayer
L1	Hardy	Jude
L2	Hardy	Well-beloved
M1	Orczy	Scarlet P
M2	Morris	News
N1	Stevenson	Ballantrae
N2	Conrad	Almayer
O1	Conrad	Lord Jim
O2	Forster	Room with view
P1	Chesterton	Man who was
P2	Forster	Room with view
Q1	Butler	Erewhon
Q2	Conrad	Almayer
R1	Chesterton	Man who was
R2	Stevenson	Catriona
S1	Morris	News
S2	Hardy	Madding
T1	Conrad	Almayer
T2	Hardy	Well-beloved
U1	Orczy	Elusive P
U2	Chesterton	Man who was
V1	Conrad	Lord Jim
V2	Forster	Room with view
W1	Orczy	Elusive P
W2	Stevenson	Catriona
X1	Hardy	Wessex Tales
X2	Hardy	Well-beloved
Y1	Tressel	Ragged TP
Y2	Orczy	Scarlet P
Z1	Tressel	Ragged TP
Z2	Hardy	Madding

Table C: Excerpts from the Brunet corpus

Id	**Author**	**Title**
1	Marivaux	La vie de Marianne
2	Marivaux	Le paysan parvenu
3	Voltaire	Zadig
4	Voltaire	Candide
5	Rousseau	La nouvelle Héloïse
6	Rousseau	Emile
7	Chateaubriand	Atala
8	Chateaubriand	La vie de Rancé
9	Balzac	Les Chouans
10	Balzac	Le cousin Pons
11	Sand	Indiana
12	Sand	La mare au diable
13	Flaubert	Madame Bovary
14	Flaubert	Bouvard et Pécuchet
15	Maupassant	Une vie
16	Maupassant	Pierre et Jean
17	Zola	Thérèse Raquin
18	Zola	La bête humaine
19	Verne	De la terre à la lune
20	Verne	Secret de Wilhelm Storitz
21	Proust	Du côté chez Swann
22	Proust	Le temps retrouvé
23	Marivaux	La vie de Marianne
24	Marivaux	Le paysan parvenu
25	Voltaire	Zadig
26	Voltaire	Candide
27	Rousseau	La nouvelle Héloïse
28	Rousseau	Emile
29	Chateaubriand	Atala
30	Chateaubriand	La vie de Rancé
31	Balzac	Les Chouans
32	Balzac	Le cousin Pons
33	Sand	Indiana
34	Sand	La mare au diable
35	Flaubert	Madame Bovary
36	Flaubert	Bouvard et Pécuchet
37	Maupassant	Une vie
38	Maupassant	Pierre et Jean
39	Zola	Thérèse Raquin
40	Zola	La bête humaine
41	Verne	De la terre à la lune
42	Verne	Secret de Wilhelm Storitz
43	Proust	Du côté chez Swann
44	Proust	Le temps retrouvé

Table D: Excerpts from the St-Jean Serie A (1-100) and St-Jean Serie B (101-200)

Id	Author	Title	Date
1	Balzac Honoré de	La cousine Bette	1846
2	Chateaubriand François-René de	Atala	1801
3	Dumas Alexandre	Le comte de Monte Cristo	1845
4	Flaubert Gustave	Bouvard et Pécuchet	1881
5	Gautier Théophile	Avatar	1856
6	Goncourt Edmond et Jules de	Madame Gervaisais	1869
7	Hugo Victor	Les Misérables	1862
8	Huysmans Joris-Karl	A rebours	1887
9	Lamartine (Alphonse de)	Graziella	1852
10	Maupassant Guy de	Bel-Ami	1885
11	Musset Alfred de	La Confession d'un enfant du siècle	1836
12	Nerval Gérard de	Aurélia	1855
13	Sand George	La Petite Fadette	1851
14	Stendhal (Henri Beyle)	La Chartreuse de Parme	1839
15	Verne Jules	De la terre à la lune	1865
16	Vigny Alfred de	Cinq-Mars	1826
17	Zola Emile	L'Argent	1891
18	Balzac Honoré de	La cousine Bette	1846
19	Chateaubriand François-René de	Atala	1801
20	Dumas Alexandre	Le comte de Monte Cristo	1845
21	Flaubert Gustave	Bouvard et Pécuchet	1881
22	Gautier Théophile	Avatar	1856
23	Goncourt Edmond et Jules de	Madame Gervaisais	1869
24	Hugo Victor	Les Misérables.	1862
25	Huysmans Joris-Karl	A rebours	1887
26	Lamartine (Alphonse de)	Graziella	1849
27	Maupassant Guy de	Bel-Ami	1885
28	Musset Alfred de	La Confession d'un enfant du siècle	1836
29	Nerval Gérard de	Aurélia	1855
30	Sand George	La Petite Fadette	1851
31	Stendhal (Henri Beyle)	La Chartreuse de Parme	1839
32	Verne Jules	De la terre à la lune	1865
33	Vigny Alfred de- Cinq-Mars	Vigny Alfred de- Cinq-Mars	1826
34	Zola Emile	L'Argent	1891
35	Balzac Honoré de	La cousine Bette	1846
36	Chateaubriand François-René de	René	1802
37	Dumas Alexandre	Le comte de Monte Cristo	1845
38	Flaubert Gustave	Madame Bovary	1857
39	Gautier Théophile	Jettatura	1856
40	Goncourt Edmond et Jules de	Germinie Lacerteux	1864
41	Hugo Victor	Les Misérables.	1862
42	Lamartine (Alphonse de)	Graziella	1852
43	Maupassant (Guy de)	Notre coeur	1890
44	Musset Alfred de	La Confession d'un enfant du siècle	1836
45	Sand George	Indiana	1832
46	Stendhal (Henri Beyle)	La Chartreuse de Parme	1839
47	Verne Jules	Le tour du monde en quatre-vingt jours	1872
48	Vigny Alfred de	Cinq-Mars	1826
49	Zola Émile	L'Assommoir	1879
50	Balzac Honoré de	César Birotteau	1837
51	Dumas Alexandre	Les trois mousquetaires	1844
52	Flaubert Gustave	Madame Bovary: moeurs de province	1857
53	Gautier Théophile	Jettatura	1856
54	Goncourt Edmond et Jules de	Germinie Lacerteux	1864
55	Hugo Victor	Notre Dame de Paris	1831
56	Maupassant (Guy de)	Notre coeur	1890
57	Sand George	Indiana	1832

58	Stendhal (Henri Beyle)	Le Rouge et le Noir	1830
59	Verne Jules	Le tour du monde en quatre-vingt jours	1872
60	Zola Émile	L'Assommoir	1879
61	Balzac Honoré de	César Birotteau	1837
62	Dumas Alexandre	Les trois mousquetaires	1844
63	Flaubert Gustave	Madame Bovary	1857
64	Gautier Théophile	Spirite	1865
65	Goncourt Edmond et Jules de	Germinie Lacerteux	1864
66	Hugo Victor	Notre Dame de Paris	1831
67	Maupassant Guy de	Fort comme la mort	1889
68	Sand George	La mare au diable	1846
69	Stendhal (Henri Beyle)	Le Rouge et le Noir -	1830
70	Vigny Alfred de	Servitude et grandeur militaires	1835
71	Zola Emile	La bête humaine	1890
72	Balzac Honoré de	Le Colonel Chabert	1832
73	Dumas Alexandre	Les trois mousquetaires	1844
74	Flaubert Gustave	Un coeur simple	1877
75	Hugo Victor	Notre Dame de Paris	1831
76	Flaubert Gustave	L'Education Sentimentale	1869
77	Maupassant Guy de	Fort comme la mort	1889
78	Sand George	La mare au diable	1846
79	Vigny Alfred de	Servitude et grandeur militaires	1835
80	Zola Emile	La bête humaine	1890
81	Balzac Honoré de	Le Colonel Chabert	1832
82	Flaubert Gustave	L'Education Sentimentale	1869
83	Maupassant Guy de	Mont Oriol	1886
84	Zola Émile	La Fortune des Rougon	1871
85	Balzac (Honoré de)	Le père Goriot	1835
86	Flaubert Gustave	Trois contes	1877
87	Maupassant Guy de	Mont Oriol	1886
88	Zola Émile	La Fortune des Rougon	1871
89	Balzac Honoré de	Eugénie Grandet	1833
90	Flaubert Gustave	Salammbô	1862
91	Maupassant Guy de	Mont Oriol	1886
92	Zola Emile	Germinal	1885
93	Balzac Honoré de	Eugénie Grandet	1833
94	Flaubert Gustave	Salammbô	1862
95	Balzac Honoré de	Le père Goriot	1835
96	Maupassant Guy de	Une vie	1883
97	Balzac Honoré de	La Maison Nucingen	1838
98	Zola Emile	Germinal	1885
99	Stendhal (Henri Beyle)	Le Rouge et le Noir	1830
100	Balzac Honoré de	La Maison Nucingen	1838
101	Barbey d'Aurevilly Jules	Le chevallier des Touches	1864
102	Bourget Paul	Une idylle tragique	1896
103	Daudet Alphonse	Le Petit Chose	1868
104	Dumas Alexandre	Le Vicomte de Bragelonne	1847
105	Staël-Holstein Anne-Louise (Madame de)	Delphine	1803
106	Erckmann Emile et Chatrian Alexandre	Histoire d'un conscrit de 1813	1864
107	France Anatole	Le crime de Sylvestre Bonnard	1881
108	Fromentin Eugène	Dominique	1862
109	Hugo Victor	Les Misérables.	1862
110	Huysmans Joris-Karl	Marthe histoire d'une fille	1876
111	Lamartine Alphonse de	Geneviève	1851
112	Loti Pierre	Pêcheur d'Islande	1886
113	Nerval Gérard de	Les Illuminés	1852
114	Proust Marcel	Les plaisirs et les jours	1896
115	Régnier Henri de	Les Rencontres de Monsieur de Bréot	1901
116	Sand Georges	La petite Fadette	1832
117	Sainte-Beuve Charles-Augustin	Volupté	1834

118	Sue Eugène	Les Mystères de Paris	1842
119	Barbey d'Aurevilly Jules	Le chevallier des Touches	1864
120	Bourget Paul	Une idylle tragique	1896
121	Daudet Alphonse	Le Petit Chose	1868
122	Dumas Alexandre	Le Vicomte de Bragelonne	1847
123	Staël-Holstein Anne-Louise (Madame de)	Delphine	1803
124	Erckmann Emile et Chatrian Alexandre	Histoire d'un conscrit de 1813	1864
125	France Anatole	Le crime de Sylvestre Bonnard	1881
126	Fromentin Eugène	Dominique	1862
127	Hugo Victor	Les Misérable	1862
128	Huysmans Joris-Karl	Marthe histoire d'une fille	1876
129	Lamartine Alphonse de	Geneviève	1851
130	Loti Pierre	Pêcheur d'Islande	1886
131	Nerval Gérard de	Les Illuminés	1852
132	Proust Marcel	Les plaisirs et les jours	1896
133	Régnier Henri de	Les Rencontres de Monsieur de Bréot	1901
134	Sand George	Indiana	1832
135	Sainte-Beuve Charles-Augustin	Volupté	1834
136	Sue Eugène	Les Mystères de Paris	1842
137	Barbey d'Aurevilly Jules	Le chevallier des Touches	1864
138	Bourget Paul	Une idylle tragique	1896
139	Daudet Alphonse	Le Petit Chose	1868
140	Dumas Alexandre	Le Vicomte de Bragelonne	1847
141	Staël-Holstein Anne-Louise (Madame de)	Delphine	1803
142	Erckmann Emile et Chatrian Alexandre	Histoire d'un conscrit de 1813	1864
143	France Anatole	Le crime de Sylvestre Bonnard	1881
144	Fromentin Eugène	Dominique	1862
145	Lamartine Alphonse de	Geneviève	1851
146	Loti Pierre	Pêcheur d'Islande	1886
147	Nerval Gérard de	Les Illuminés	1852
148	Proust Marcel	Les plaisirs et les jours	1896
149	Régnier Henri de	Les Rencontres de Monsieur de Bréot	1901
150	Sand George	La Petite Fadette	1851
151	Sainte-Beuve Charles-Augustin	Volupté	1834
152	Sue Eugène	Les Mystères de Paris	1842
153	Barbey d'Aurevilly Jules	Le chevallier des Touches	1864
154	Bourget Paul	Une idylle tragique	1896
155	Daudet Alphonse	Le Petit Chose	1868
156	Dumas Alexandre	Le Vicomte de Bragelonne	1847
157	Staël-Holstein Anne-Louise (Madame de)	Delphine	1803
158	Erckmann Emile et Chatrian Alexandre	Histoire d'un conscrit de 1813	1864
159	France Anatole	Le crime de Sylvestre Bonnard	1881
160	Fromentin Eugène	Dominique	1862
161	Lamartine Alphonse de	Geneviève	1851
162	Loti Pierre	Pêcheur d'Islande	1886
163	Régnier Henri de	Les Rencontres de Monsieur de Bréot	1901
164	Sand George	Indiana	1832
165	Sainte-Beuve Charles-Augustin	Volupté	1834
166	Sue Eugène	Les Mystères de Paris	1842
167	Barbey d'Aurevilly Jules	Les Diabolique	1874
168	Bourget Paul	Une idylle tragique	1896
169	Daudet Alphonse	Le Petit Chose	1868
170	France Anatole	La Rôtisserie de la reine Pédauque	1893
171	Fromentin Eugène	Dominique	1862
172	Loti Pierre	Pêcheur d'Islande	1886
173	Régnier Henri de	La Double Maîtresse	1900
174	Sue Eugène	Les Mystères de Paris	1842
175	Barbey d'Aurevilly Jules	Les Diabolique	1874
176	Bourget Paul	Une idylle tragique	1896
177	Daudet Alphonse	Le Petit Chose	1868
178	France Anatole	La Rôtisserie de la reine Pédauque	1893

179	Loti Pierre	Madame Chrysanthème	1899
180	Régnier Henri de	La Double Maîtresse	1900
181	Sainte-Beuve Charles-Augustin	Volupté	1834
182	Sue Eugène	Les Mystères de Paris	1842
183	Staël-Holstein Anne-Louise (Madame de)	Delphine	1803
184	Barbey d'Aurevilly Jules	Les Diaboliques	1874
185	Vallès Jules	L'Enfant	1879
186	France Anatole	La Rôtisserie de la reine Pédauque	1893
187	Loti Pierre	Madame Chrysanthème	1899
188	Régnier Henri de	La Double Maîtresse	1900
189	Sainte-Beuve Charles-Augustin	Volupté	1834
190	Sue Eugène	Les Mystères de Paris	1842
191	Staël-Holstein Anne-Louise (Madame de)	Delphine	1803
192	Barbey d'Aurevilly Jules	Les Diaboliques	1874
193	Vallès Jules	L'Enfant	1879
194	Régnier Henri de	La Double Maîtresse	1900
195	Sue Eugène	Les Mystères de Paris	1842
196	France Anatole	La Rôtisserie de la reine Pédauque	1893
197	Vallès Jules	L'Enfant	1879
198	Sue Eugène	Les Mystères de Paris	1842
199	Vallès Jules	L'Enfant	1879
200	Sue Eugène	Les Mystères de Paris	1842

Individual Rank Lists Methods

Table E: MF Tokens occurrences in St-Jean A and B

Rank	Token	Occurrences
1	,	167892
2	de	82443
3	.	70406
4	et	49195
5	la	47291
6	le	38975
7	à	37913
8	il	33207
9	l’	30639
10	les	28510
11	un	26157
12	d’	24886
13	du	24275
14	que	24060
15	je	23274
16	en	23238
17	une	19997
18	au	19137
19	des	18854
20	elle	18772
21	qui	18534
22	-	16861
23	;	16637
24	dans	16322
25	qu’	16245
26	ne	16162
27	!	15900
28	vous	15016
29	pas	14531
30	ce	13633
31	se	13310
32	est	13222
33	pour	12479
34	s’	11910
35	n’	11556
36	son	11294
37	était	10949
38	lui	10704
39	plus	10554
40	avait	9851

Table F: MF Lemma occurrences in St-Jean A and B

Rank	Lemma	Occurrences
1	le	169563
2	,	167892
3	de	138448
4	.	70406
5	il	51978
6	à	51191
7	et	49195
8	je	46782
9	un	46331
10	être	43999
11	que	40309
12	avoir	36548
13	ce	34084
14	son	27923
15	ne	27714
16	se	24075
17	en	23235
18	qui	18534
19	-	16861
20	;	16637
21	dans	16324
22	!	15900
23	vous	15034
24	pas	14532
25	tout	13982
26	pour	12484
27	mon	12188
28	dire	11489
29	faire	10933
30	lui	10702
31	plus	10553
32	...	9779
33	“	9705
34	sur	9391
35	comme	8754
36	avec	8707
37	si	8569
38	mais	8528
39	on	8520
40	?	8471

Table G: Average precisions for n-MF Tokens and Lemmas on St-Jean

(a) Tokens

Distance metric	$n = 250$	$n = 500$	$n = 750$	$n = 1000$	$n = 1250$	$n = 1500$	$n = 1750$	$n = 2000$
Manhattan	0.73	0.76	0.76	0.76	0.74	0.72	0.69	0.68
Tanimoto	0.72	0.74	0.75	0.75	0.74	0.75	0.74	0.74
Euclidean	0.69	0.67	0.64	0.64	0.58	0.55	0.52	0.49
Matusita	0.72	0.74	0.73	0.73	0.71	0.71	0.70	0.70
Clark	0.55	0.71	0.74	0.76	0.76	0.76	0.75	0.74
Cosine distance	0.76	0.79	0.78	0.79	0.78	0.78	0.78	0.78
Kld	0.67	0.70	0.69	0.68	0.65	0.64	0.64	0.63
J-Divergence	0.71	0.72	0.71	0.71	0.68	0.68	0.67	0.66

(b) Lemmas

Distance metric	$n = 250$	$n = 500$	$n = 750$	$n = 1000$	$n = 1250$	$n = 1500$	$n = 1750$	$n = 2000$
Manhattan	0.75	0.76	0.77	0.73	0.69	0.62	0.58	0.54
Tanimoto	0.73	0.75	0.76	0.76	0.76	0.75	0.75	0.75
Euclidean	0.70	0.66	0.63	0.59	0.55	0.50	0.48	0.44
Matusita	0.73	0.75	0.75	0.75	0.73	0.72	0.71	0.70
Clark	0.56	0.67	0.73	0.73	0.72	0.68	0.65	0.62
Cosine distance	0.79	0.80	0.79	0.78	0.78	0.77	0.77	0.76
Kld	0.69	0.71	0.71	0.69	0.67	0.65	0.65	0.63
J-Divergence	0.72	0.73	0.73	0.72	0.70	0.68	0.67	0.66

Table H: St-Jean rank list degradation over the text size. Rank list computed using text representation 0

Number of tokens	AP	RPrec	HPrec
9000	0.75	0.67	189.00
8750	0.75	0.67	188.00
8500	0.75	0.67	185.00
8250	0.75	0.67	181.00
8000	0.74	0.66	195.00
7750	0.72	0.65	180.00
7500	0.72	0.64	158.00
7250	0.71	0.62	166.00
7000	0.70	0.61	166.00
6750	0.69	0.61	166.00
6500	0.67	0.60	184.00
6250	0.67	0.59	153.00
6000	0.66	0.60	188.00
5750	0.65	0.58	171.00
5500	0.65	0.58	197.00
5250	0.63	0.55	178.00
5000	0.62	0.56	143.00
4750	0.61	0.56	99.00
4500	0.61	0.54	107.00
4250	0.59	0.54	98.00
4000	0.57	0.52	61.00
3750	0.55	0.51	60.00
3500	0.53	0.48	70.00
3250	0.51	0.47	97.00
3000	0.49	0.46	90.00
2750	0.47	0.44	72.00
2500	0.44	0.42	51.00
2250	0.42	0.40	37.00
2000	0.39	0.38	27.00
1750	0.37	0.37	41.00
1500	0.35	0.36	42.00
1250	0.29	0.32	17.00
1000	0.26	0.29	33.00
750	0.21	0.25	13.00
500	0.14	0.19	7.00
250	0.08	0.12	3.00

Table I: MF 4-grams occurrences in St-Jean A and B

Rank	4-grams	occurrences
1	_the	39061
2	the_	30122
3	and_	18395
4	_and	16771
5	_of_	16221
6	ing_	14375
7	_to_	13566
8	hat_	8973
9	_he_	8892
10	_._.	8545
11	._._	8545
12	_in_	8480
13	_tha	8137
14	his_	7927
15	_was	7569
16	her_	7554
17	was_	7491
18	that	7128
19	_you	6933
20	n_th	6852
21	_it_	6233
22	d_th	6210
23	ther	5826
24	_his	5697
25	t_th	5676
26	_,_a	5646
27	_for	5632
28	ere_	5617
29	_not	5616
30	f_th	5470
31	of_t	5340
32	he_s	5296
33	you_	5198
34	e_th	5171
35	_wit	5165
36	_had	5159
37	had_	5150
38	_her	5102
39	with	5101
40	not_	4961

Table J: MF 5-grams occurrences in St-Jean A and B

Rank	5-grams	occurrences
1	_the_	30072
2	_and_	16693
3	_._._	8545
4	_was_	7473
5	_that	7128
6	that_	7115
7	._._.	7049
8	_his_	5631
9	n_the	5612
10	_of_t	5331
11	_you_	5197
12	_had_	5150
13	_with	5067
14	of_th	5031
15	_not_	4814
16	f_the	4722
17	d_the	4495
18	with_	4380
19	_for_	4315
20	_,_an	4295
21	,_and	4256
22	t_the	4148
23	ould_	4039
24	_her_	4024
25	_she_	3811
26	e_of_	3636
27	the_s	3602
28	here_	3459
29	_but_	3430
30	_._”_	3274
31	in_th	3219
32	ther_	3200
33	_in_t	3170
34	and_t	3079
35	said_	2935
36	nd_th	2930
37	_said	2912
38	e_the	2884
39	_him_	2871
40	have_	2769

Table K: Individual methods summary: Oxquarry and Brunet

(a) Oxquarry

TR ID	AP	RPrec	HPrec
0	0.89	0.79	92
1	0.89	0.80	88
2	0.67	0.56	39
3	0.63	0.57	54
4	0.77	0.68	61
5	0.78	0.68	84
6	0.79	0.69	74

(b) Brunet

TR ID	AP	RPrec	HPrec
0	0.72	0.64	19
1	0.71	0.67	8
2	0.68	0.62	19
3	0.67	0.64	22
4	0.73	0.67	24
5	0.74	0.67	20
6	0.76	0.70	25

Table L: Individual methods summary: St-Jean

(a) St-Jean A

TR ID	AP	RPrec	HPrec
0	0.78	0.69	65
1	0.75	0.66	68
2	0.77	0.70	99
3	0.76	0.69	80
4	0.71	0.61	75
5	0.76	0.65	72
6	0.73	0.63	83
7	0.68	0.63	20
8	0.77	0.69	75

(b) St-Jean B

TR ID	AP	RPrec	HPrec
0	0.94	0.87	165
1	0.91	0.81	124
2	0.93	0.86	156
3	0.89	0.82	169
4	0.91	0.83	172
5	0.92	0.86	182
6	0.87	0.77	153
7	0.91	0.83	132
8	0.88	0.83	141

(c) St-Jean A and B

TR ID	AP	RPrec	HPrec
0	0.78	0.70	253
1	0.74	0.66	175
2	0.76	0.70	203
3	0.75	0.69	204
4	0.74	0.64	231
5	0.78	0.68	251
6	0.70	0.62	219
7	0.73	0.65	60
8	0.75	0.69	185

Authorship Clustering

Table M: Silhouette-based clustering evaluation, $\alpha = 0.0$

Rank list		Linkage criterion		
TR ID	Corpus	Single	Average	Complete
0	Oxquarry	0.84/0.08	0.84/0.08	0.80/0.10
1	Oxquarry	0.79/0.12	0.79/0.12	0.76/0.15
2	Oxquarry	0.62/0.42	0.74/0.13	0.80/0.12
3	Oxquarry	0.72/0.25	0.75/0.19	0.73/0.23
4	Oxquarry	0.77/0.17	0.80/0.10	0.80/0.10
5	Oxquarry	0.80/0.12	0.80/0.10	0.80/0.10
6	Oxquarry	0.80/0.10	0.80/0.10	0.75/0.12
0	Brunet	0.67/0.27	0.69/0.27	0.70/0.25
1	Brunet	0.77/0.20	0.76/0.23	0.76/0.23
2	Brunet	0.65/0.36	0.72/0.27	0.77/0.20
3	Brunet	0.65/0.36	0.72/0.27	0.73/0.23
4	Brunet	0.69/0.27	0.69/0.27	0.71/0.23
5	Brunet	0.68/0.25	0.69/0.27	0.70/0.25
6	Brunet	0.72/0.25	0.73/0.23	0.74/0.20
0	St-Jean A	0.61/0.31	0.66/0.21	0.60/0.26
1	St-Jean A	0.57/0.37	0.59/0.30	0.57/0.31
2	St-Jean A	0.64/0.31	0.64/0.23	0.61/0.25
3	St-Jean A	0.51/0.39	0.64/0.23	0.55/0.32
4	St-Jean A	0.60/0.27	0.57/0.31	0.60/0.26
5	St-Jean A	0.57/0.32	0.58/0.30	0.61/0.25
6	St-Jean A	0.56/0.32	0.60/0.27	0.65/0.22
7	St-Jean A	0.57/0.41	0.72/0.16	0.69/0.18
8	St-Jean A	0.66/0.29	0.73/0.19	0.58/0.29
0	St-Jean B	0.93/0.06	0.89/0.08	0.90/0.07
1	St-Jean B	0.85/0.16	0.90/0.05	0.88/0.07
2	St-Jean B	0.94/0.05	0.98/0.02	0.91/0.05
3	St-Jean B	0.94/0.05	0.90/0.06	0.91/0.05
4	St-Jean B	0.91/0.07	0.87/0.09	0.90/0.06
5	St-Jean B	0.91/0.07	0.89/0.08	0.92/0.05
6	St-Jean B	0.87/0.10	0.90/0.06	0.91/0.05
7	St-Jean B	0.90/0.08	0.88/0.09	0.87/0.08
8	St-Jean B	0.90/0.09	0.95/0.04	0.93/0.02

Table N: Silhouette-based clustering evaluation, $\alpha = -0.2$, $B^3_{F_1}/r_{diff}$

Rank list		Linkage criterion		
TR ID	Corpus	Single	Average	Complete
0	Oxquarry	0.95/0.02	0.97/-0.02	0.84/0.00
1	Oxquarry	0.89/0.02	0.93/0.00	0.81/0.02
2	Oxquarry	0.62/0.27	0.34/-0.13	0.74/0.06
3	Oxquarry	0.70/0.06	0.72/0.04	0.77/0.02
4	Oxquarry	0.87/0.02	0.87/0.02	0.80/0.02
5	Oxquarry	0.84/0.02	0.87/0.02	0.80/0.02
6	Oxquarry	0.80/0.02	0.76/0.00	0.76/0.00
0	Brunet	0.78/0.11	0.82/0.11	0.73/0.16
1	Brunet	0.81/0.09	0.85/0.11	0.84/0.07
2	Brunet	0.74/0.14	0.75/0.02	0.74/0.05
3	Brunet	0.78/0.05	0.80/0.09	0.78/0.11
4	Brunet	0.75/0.14	0.78/0.09	0.80/0.14
5	Brunet	0.78/0.09	0.82/0.11	0.87/0.11
6	Brunet	0.85/0.09	0.79/0.09	0.85/0.09
0	St-Jean A	0.69/0.13	0.75/0.10	0.69/0.14
1	St-Jean A	0.66/0.22	0.71/0.13	0.71/0.12
2	St-Jean A	0.73/0.14	0.85/0.02	0.88/0.04
3	St-Jean A	0.75/0.11	0.79/0.05	0.80/0.09
4	St-Jean A	0.65/0.17	0.70/0.13	0.71/0.13
5	St-Jean A	0.70/0.14	0.72/0.13	0.73/0.12
6	St-Jean A	0.74/0.15	0.85/0.08	0.82/0.10
7	St-Jean A	0.67/0.17	0.81/0.02	0.73/0.00
8	St-Jean A	0.74/0.05	0.76/-0.02	0.79/0.04
0	St-Jean B	0.89/0.00	0.93/-0.02	0.89/-0.01
1	St-Jean B	0.87/0.03	0.86/-0.04	0.88/-0.04
2	St-Jean B	0.83/-0.02	0.83/-0.04	0.85/-0.04
3	St-Jean B	0.87/-0.03	0.87/-0.02	0.80/-0.02
4	St-Jean B	0.88/0.03	0.91/-0.01	0.89/-0.02
5	St-Jean B	0.88/0.00	0.91/-0.03	0.91/-0.03
6	St-Jean B	0.82/0.01	0.89/0.00	0.89/-0.03
7	St-Jean B	0.86/0.01	0.88/-0.05	0.83/-0.03
8	St-Jean B	0.87/0.03	0.80/-0.04	0.80/-0.04

Rank List Fusions

Table O: Fusions evaluation of every retained rank lists for each dataset, Metrics: AP/RPrec/HPrec

(a) Z-Score fusion

Corpus	Metrics
Oxquarry	0.84/0.74/90
Brunet	0.76/0.70/21
St-Jean A	0.85/0.77/95
St-Jean B	0.96/0.90/200
Mean	0.85/0.78/101.50

(b) Regression fusion for each pair training/testing

		Training				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Testing	Oxquarry	0.87/0.79/92	0.85/0.76/90	0.86/0.76/89	0.85/0.76/89	0.86/0.77/90.00
	Brunet	0.76/0.70/20	0.76/0.70/20	0.76/0.70/22	0.76/0.71/21	0.76/0.70/20.75
	St-Jean A	0.84/0.75/90	0.84/0.75/85	0.85/0.75/102	0.84/0.74/101	0.84/0.75/94.50
	St-Jean B	0.95/0.89/182	0.95/0.90/180	0.96/0.91/196	0.96/0.91/197	0.96/0.90/188.75
	Mean	0.86/0.78/96.00	0.85/0.78/93.75	0.86/0.78/102.25	0.85/0.78/102.00	0.85/0.78/98.50

Fusions and Clustering

Table P: Silhouette-based clustering evaluation on Z-Scores rank lists, $B^3_{F_1}/r_{diff}$, Maximal Silhouette ($\alpha = 0$)

	Linkage criterion		
Corpus	Single	Average	Complete
Oxquarry	0.80/0.10	0.80/0.10	0.80/0.10
Brunet	0.67/0.32	0.72/0.20	0.70/0.25
St-Jean A	0.57/0.30	0.57/0.29	0.60/0.25
St-Jean B	0.89/0.08	0.91/0.05	0.91/0.05
Absolute mean	0.73/0.20	0.75/0.16	0.75/0.16

Table Q: Silhouette-based clustering evaluation on Z-Scores rank lists, $B^3_{F_1}/r_{diff}$, $\alpha = -0.2$

	Linkage criterion		
Corpus	Single	Average	Complete
Oxquarry	0.85/ 0.02	0.96/ 0.02	0.87/ 0.04
Brunet	0.82/ 0.05	0.85/ 0.09	0.82/ 0.11
St-Jean A	0.77/ 0.13	0.79/ 0.12	0.74/ 0.12
St-Jean B	0.92/-0.01	0.89/-0.03	0.91/-0.03
Absolute mean	0.84/0.05	0.87/0.07	0.84/0.08

Table R: Distribution-based clustering evaluation, Z-Score rank lists $B^3_{F_1}/r_{diff}$ for each corpus pair

(a) Average Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.85/-0.04	0.71/-0.07	0.64/-0.09	0.72/-0.10	0.73/-0.07
	Brunet	0.96/ 0.02	0.81/ 0.00	0.75/-0.07	0.82/-0.06	0.84/-0.03
	St-Jean A	0.89/ 0.04	0.81/ 0.00	0.86/-0.04	0.92/-0.02	0.87/-0.01
	St-Jean B	0.82/ 0.08	0.82/ 0.07	0.88/-0.03	0.95/-0.01	0.87/ 0.03
	Mean	0.88/ 0.02	0.79/ 0.00	0.78/-0.06	0.85/-0.05	0.83/-0.02

(b) Complete Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.82/ 0.08	0.81/ 0.07	0.86/-0.02	0.91/-0.03	0.85/0.02
	Brunet	0.82/ 0.08	0.81/ 0.07	0.87/-0.01	0.92/-0.02	0.86/0.03
	St-Jean A	0.82/ 0.08	0.85/ 0.09	0.87/-0.01	0.97/ 0.00	0.88/0.04
	St-Jean B	0.80/ 0.10	0.84/ 0.14	0.84/ 0.04	0.90/ 0.04	0.85/0.08
	Mean	0.82/ 0.08	0.83/ 0.09	0.86/ 0.00	0.93/-0.00	0.86/0.04

Table S: Regression-based clustering evaluation, Z-Score rank lists $B^3_{F_1}/r_{diff}$ for each corpus pair

(a) Average Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.82/ 0.08	0.82/ 0.07	0.87/-0.02	0.95/-0.01	0.86/0.03
	Brunet	0.80/ 0.10	0.75/ 0.18	0.82/ 0.07	0.91/ 0.05	0.82/0.10
	St-Jean A	0.80/ 0.10	0.82/ 0.11	0.84/ 0.04	0.95/ 0.03	0.85/0.07
	St-Jean B	0.80/ 0.10	0.76/ 0.16	0.83/ 0.05	0.95/ 0.03	0.84/0.08
	Mean	0.81/ 0.09	0.79/ 0.13	0.84/ 0.04	0.94/ 0.03	0.84/0.07

(b) Complete Linkage

		Testing				
		Oxquarry	Brunet	St-Jean A	St-Jean B	Mean
Training	Oxquarry	0.80/0.10	0.84/0.14	0.83/0.05	0.91/0.05	0.85/0.08
	Brunet	0.73/0.19	0.77/0.20	0.67/0.19	0.89/0.07	0.77/0.16
	St-Jean A	0.78/0.13	0.78/0.18	0.74/0.14	0.91/0.05	0.80/0.13
	St-Jean B	0.78/0.13	0.78/0.18	0.70/0.16	0.90/0.06	0.79/0.13
	Mean	0.77/0.14	0.79/0.18	0.74/0.14	0.90/0.06	0.80/0.13