Automatic language processing applied to the militarization of children's literature from 1860 to 1919

Samuel Gonçalves
(supervisor: Fabrice Boissier, Marie Puren)

Technical Report n^o202306-techrep-goncalves, June 2023
revision c636a38

This report focuses on the analysis of language using the CREA method and its comparison with other methods applied to the militarization of youth literature from 1860 to 1919.

The project aims to address the research question of whether the militarization of children's literature during World War I was a direct consequence of the conflict or a manifestation of a preexisting trend since 1860.

The project's objective is to extract the vocabulary and lexical field that indicate the strength of the militarization trend from the selected texts.

The report describes the development of tools for document retrieval, tokenization, lemmatization, application of the CREA method, and visualization of results.

Ce rapport porte sur l'analyse de la langue à l'aide de la méthode CREA et sa comparaison avec d'autres méthodes appliquées à la militarisation de la littérature pour la jeunesse de 1860 à 1919.

Le projet vise à répondre à la question de recherche de savoir si la militarisation de la littérature de jeunesse pendant la Première Guerre mondiale était une conséquence directe du conflit ou une manifestation d'une tendance préexistante depuis 1860.

L'objectif du projet est d'extraire des textes sélectionnés le vocabulaire et le champ lexical qui indiquent la force de la tendance à la militarisation.

Le rapport décrit le développement d'outils pour la recherche de documents, la tokenisation, la lemmatisation, l'application de la méthode CREA et la visualisation des résultats.

Keywords
Analysis of language, CREA method, militarization, youth literature, children's literature, World War I, document retrieval, tokenization, lemmatization, visualization

Laboratoire de Recherche de l'EPITA
14-16, rue Voltaire – FR-94276 Le Kremlin-Bicêtre CEDEX – France
Tél. +33 1 53 14 59 22 – Fax. +33 1 53 14 59 13
samuel.goncalves@epita.fr – http://www.lre.epita.fr/

Copying this document

Contents

1 Introduction **4**

2 Context and state of the art **6**

3 Chain of tools for processing large volumes of historical data **8**
3.1 Scrapping . . . 8
3.2 OCRization . . . 9
3.3 Pre-Processing . . . 10
3.3.1 Merging . . . 10
3.3.2 Tokenization . . . 10
3.3.3 Lemmatization . . . 10
3.3.4 Babelization . . . 11
3.4 Processing . . . 13
3.4.1 Filtering and calculation of the number of occurrences . . . 13
3.4.2 Normalization . . . 13
3.4.3 Formal context creation . . . 14
3.4.4 Analysis of the formal context . . . 14
3.5 Visualization . . . 16

4 Measures **18**
4.1 Pre-Processing . . . 18
4.2 Visualization . . . 19

5 Discussion and Conclusion **20**
5.1 Discussion . . . 20
5.1.1 Related Work . . . 20
5.1.2 Future Work . . . 20
5.2 Conclusion . . . 20

6 Bibliography **21**

Chapter 1

Introduction

The growth in the number of digitized legacy works since the 1990s has opened up new perspectives for historical research Brosset (2016). The exploitation of these large volumes of digitized documents opens up the possibility of using a wide range of computer techniques to analyze these texts. For example, Gallica, the digital library of the Bibliothèque nationale de France, makes available digitized and OCRized texts, which can be collected via APIs. Our project involves analyzing documents collected via Gallica's APIs using computational methods used for knowledge extraction and text mining. The originality of the project lies in the fact that we are working closely with a teacher-researcher in contemporary history, using computer science to answer a historical research question. Another aspect of the project is to create scripts and develop a tool that can be used in the future to study other digitized ancient texts.

The research question our work will answer is: was the militarization of children's literature during the First World War one of the direct effects of the conflict, or was it the exacerbation of an underlying trend already present in France since 1870? The First World War was a pivotal moment for children's literature, which became politicized and militarized as a result of intensive propaganda aimed at the civilian population - and children in particular Audouin-Rouzeau (1993). To answer this question, we use documents from Gallica's Children's Literature collections, with a particular focus on the children's press. We hypothesized that the children's press, with its serial nature and ability to appeal to a wide readership, should give a good idea of the "trends" themes in children's literature prior to the outbreak of the Great War. The documents used are post-1870, on the assumption that the trauma of the 1870 defeat was the main catalyst for the militarization of children's literature. We contrast this corpus of texts with a major collection of novels published during the First World War by Larousse, "Les Livres Roses de la guerre". The latter is available on Gallica and crystallizes most of the major themes and motifs used in propaganda literature aimed at young people Puren (2013, 2016). The RDI project aims to extract the vocabulary and the entire lexical field that can demonstrate the strength of the trend from the selected texts.

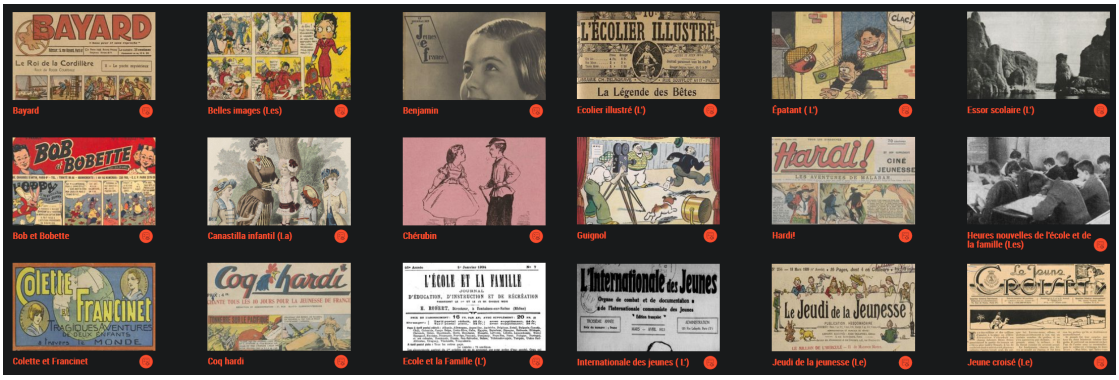


Figure 1.1: Overview of texts in Gallica's "children's press" collection

The work presented here has resulted in the creation of several tools: a tool for retrieving documents from the digital library of the Bibliothèque nationale de France, ordered by year and by month in a file tree (3.1), a tokenization and lemmatization tool adapted to the processing of a database with a temporal dimension to prepare it for the application of algorithms such as LDA (3.3), a tool that applies the CREA method to a database with a temporal dimension (3.4) and a tool for visualizing the results of the CREA method which does not require the use of Gephi (3.5). As the application of these tools to the database is not yet complete, it is not yet possible to answer the research question.

The work is composed of four main parts leading to the tools presented.

- The retrieval of a database from the Bibliothèque nationale de France.
- The database processing as preparation for LDA.
- The application of the CREA method.
- The visualization of the resulting metrics of the CREA method.

Chapter 2

Context and state of the art

To find out whether the militarization of children's literature between 1870 and 1919 was the result of underlying trends or of an acceleration due to the First World War, we need to look at military vocabulary and its evolution over a pre-established corpus of texts. The specification of what constitutes military vocabulary also needs to be taken into account: here, it is taken in the sense of language analysis. The words belonging to the military vocabulary within our corpus will therefore be defined by clustering Everitt et al. (2011) the texts.

The project therefore requires the application of pre-existing clustering algorithms, with a view to comparing the accuracy and relevance of their results (4). Topic formation using the LDA method Blei et al. (2003) and clustering using conceptual similarity graphs produced by the CREA method Boissier (2022) will be compared. The creation and analysis of mutual impact graphs produced by the CREA method per time slice studied, correlated with the military cluster, will serve as a result of the research question.

Application of these methods requires preparation of the text corpus. This can be divided into several stages.

- Preparation of documents for the LDA method Blei et al. (2003):
  - Document tokenization (3.3.2, 3.3.4). In the context of language analysis and the Natural Language Toolkit Bird (2006), a token is a string of characters between two spaces, whether one word or several, linked by a character such as a hyphen or apostrophe. Babelfy's Moro et al. (2014) application to the Latent Dirichlet Allocation Ekinci and İlhan Omurca (2020) handles tokenization using the classic decomposition of identifiers in the BabelNet semantic network, with tokens representing concepts or named entities.
  - Token lemmatization (3.3.3). Lemmatization is a lexical treatment of tokens that returns them to a canonical neutral form. Snowball Porter (2001) sees lemmatization as stemming: it applies a desuffixation to tokens from a list of suffixes known per language. TreeTagger Schmid (1994, 1995) doesn't take this approach, seeking to return to the masculine singular and the indicative for verbs. Babelfy's application of Latent Dirichlet Allocation does not require lemmatization in that identifiers in the BabelNet network are equivalent to lemmas: two words of common meaning will be represented by the same identifier.

- Preparing documents for the CREA method Boissier (2022):
  - The analysis of the formal concept Wille (1982) of the documents determined after a calculation of the number of occurrences (3.4.1), a normalization (3.4.2), and the application of the high or direct strategy Jaffal (2019) in the creation of the formal concept (3.4.3) Belohlavek (2008).
  - Use of the Galois lattice Wille (1982) to calculate CREA metrics Boissier (2022), mutual impact and conceptual similarity (3.4.4).

These preparations require the retrieval of the text corpus (3.1), which must be semi-automated given the size of the corpus. To achieve this, the Gallica API must be used. Pyllica was used for this project, enabling requests to be made to this API from Python.

This project also relies on the OCRization of documents (3.2), as the OCRization provided by Gallica is not satisfactory for our needs. This OCRization work will not be dealt with in detail here.

Finally, the results obtained should be presented in the form of a visualization (3.5). Here, F. Boissier's visualizations Boissier (2022) using ForceAtlas and Gephi Bastian et al. (2009) will be compared with the results obtained using NetworkX Hagberg et al. (2008) and the Fruchterman-Reingold force-directed algorithm Fruchterman and Reingold (1991).

As far as time slices are concerned, the choice was made to store documents in a tree structure by year and month, to enable slices of varying long-term precision. For the application of the tool within the framework of the project and the research question, time slices of one year were chosen, a good compromise between the need to have a certain number of documents per slice to obtain exploitable results and a hardware constraint in terms of memory capacity and time for the execution of the chosen algorithms.

Chapter 3

Chain of tools for processing large volumes of historical data

3.1 Scrapping

Scrapping is used to semi-automatically download documents from Gallica, a collection of works from the Bibliothèque nationale de France, using Pyllica.

Input		Output
Bibliothèque nationale de France	→	pdf files

(a) Bibliothèque nationale de France.

(b) Gallica.

Pyllica is a Python-based tool for retrieving documents from the Gallica digital library. In particular, it can be used to rapidly build up large corpora for computer-assisted analysis. This library simplifies the task of querying the Gallica API, but suffers from certain problems.

- Each item retrieved is a different server query, and there is a limit on the number of queries per person per minute. We therefore need to use a method that allows us to wait a certain amount of time between each call to *textpress*, the document retrieval function. This problem can easily be solved by setting up a cron job (a program that automatically executes a script at a pre-specified date and time).
- The *rate* option in the document retrieval function lets you select the time in days between two publications. Problems with this option are:
  - Releases based on month and not day ("Every first Monday of the month", "Every 1st of the month").

  - Exceptions (one resource is published every 1st of the month, but sometimes special issues are published on the 15th).

These problems have been solved by the external addition of the ability to select the time in months between 2 publications, and by manual management of exceptions. The easiest way to retrieve all resources would be to run the retrieval on all days in the selected period, from January 1, 1860 to December 31, 1919. Unfortunately, this method would require 335 days to retrieve all the desired documents. The selection must therefore be made manually. This is why the retrieval is considered semi-automatic in the sense that, although the execution of the queries is automated, the selection of the database contents is not. It takes the form of a file made up of groups of 10 lines:

- Document name (e.g. La Jeunesse moderne)
- Document link (e.g. https://gallica.bnf.fr/ark:/12148/cb32796317q/date)
- The year of the first document to be retrieved (ex: 1904)
- The month of the first document to be retrieved (ex: 10)
- The day of the first document to be retrieved (ex: 1)
- The year of the last document to be recovered (ex: 1910)
- The month of the last document to be recovered (ex: 1)
- The day of the last document to be retrieved (ex: 22)
- Time between two successive documents (ex: 7)
- Time unit (d or m)

3.2 OCRization

Optical character recognition is used to convert images into machine-encoded text.

Input		Output
pdf files	→	folder containing one text file per page

As indicated in the context (2), the OCRization will not be detailed here.

3.3 Pre-Processing

Pre-processing aim to eliminate OCR errors and to apply tokenization and lemmatization processing Schmid (1994, 1995), or other tools Moro et al. (2014) to extract useful vocabulary.

3.3.1 Merging

Input		Output
folder containing one text file per page	→	txt files

At the end of the OCRization, a document is represented by a folder containing one page per file. The first two pages contain the Gallica logo and terms of use. When pages are merged, these ones are removed. At the same time, OCR noise characters such as "@" are also removed.

3.3.2 Tokenization

Input		Output
txt files	→	tokens

At this stage, the text appears as normal, with sentences, line breaks and spaces. This stage consists in transforming the text so that only one token is kept per line. Since tokenization is part of a chain whose purpose is to apply LDA, the ultimate aim is to make the representation of similar words identical. Thus, tokens that do not provide information on meaning (such as determiners and articles) are also removed at the tokenization stage.

3.3.3 Lemmatization

Input		Output
tokens	→	lemmas

Lemmatization via Snowball consists of a return to the root. Snowball is contained in python's NLTK library. The idea behind this stemmer is to consider that in French, different representations of similar words are mainly due to:

- gender agreement (often represented by the addition of an e).
- number agreement (often represented by the addition of an s or an x).
- conjugation (which can take different forms depending on the tense and person).

Snowball searches for the suffixes mentioned above, removing them (de-suffixing) to proceed with lemmatization.

Lemmatization via TreeTagger consists of returning masculine singular, or indicative verbs. The algorithm used is more complex, allowing us to handle cases not handled by Snowball (4.1).

The lemmatization algorithm simply applies one of these two methods to the set of tokens retrieved from the text corpus, after setting the language to french.

3.3.4 Babelization

Input		Output
txt files	→	babel-tokens

Babelization is the name given to the application of Babelfy's API to the lexical disambiguation of a text's semantic units. Unlike the semantic units of tokenization and lemmatization in the classical method, where the tokens were words, via Babelfy these semantic units are concepts and named entities. This means that they can be words, but also groups of words, if these groups are seen as more decisive in their meaning than the words that make them up separately.

Figure 3.2: Illustration of Babelfy's application on a french sentence.

The Babelfy API can be accessed via various Python libraries, notably pybabelfy and BabelPy. As both are equivalent, pybabelfy is arbitrarily chosen in this project.

Requests to the Babelfy API are made in the form of a URI, limiting the number of characters that can be given within a request. The work therefore involved setting up an algorithm for calculating text decomposition into a list of index pairs (start and end) for each block, based on an average text block size and a maximum deviation from this value in the form of a percentage.

The babel-token is the name given to the equivalent of the token in the analogy between the CREA method and the LDA. It is the basic brick containing all the information for applying the CREA method. It breaks down into 7 parts.

- The index of the first character of the content within the text.
- The index of the last character of the content within the text.
- The text content between the two indexes.
- An identifier in the Babelnet semantic network, representing a concept or named entity corresponding to the content.
- The "score", without qualifier.
- The "global score".
- The "consistency score".

These different scores provide additional information on the relevance of finding this concept or named entity within this text (Babelfy's method being contextual), the relevance of finding this concept or named entity written in this form (errors due to OCR noise, for example), and the relevance of finding this concept or named entity written in this location (agreement or conjugation errors).

Babelization can be seen as tokenization + lemmatization if only the identifiers are kept. Indeed, by definition of the semantic network, two similar words will have the same identifier, so they will be represented by the same token. Thus, via this representation (called "equivalent text"), babelization is acceptable as an input block in the processing chain leading to LDA. However, there is still a problem in measuring the results obtained. Babelfy identifiers do not provide a satisfactory way of knowing, as a human, whether a specific term should belong to a specific cluster. There are two ways of solving this problem.

- Each time a new id is added to the list of known identifiers, by keeping the equivalent string in memory so as to have a reference word/example enabling a human eye to understand the results.
- By referring to the Babelnet semantic network for indications concerning the identifier.

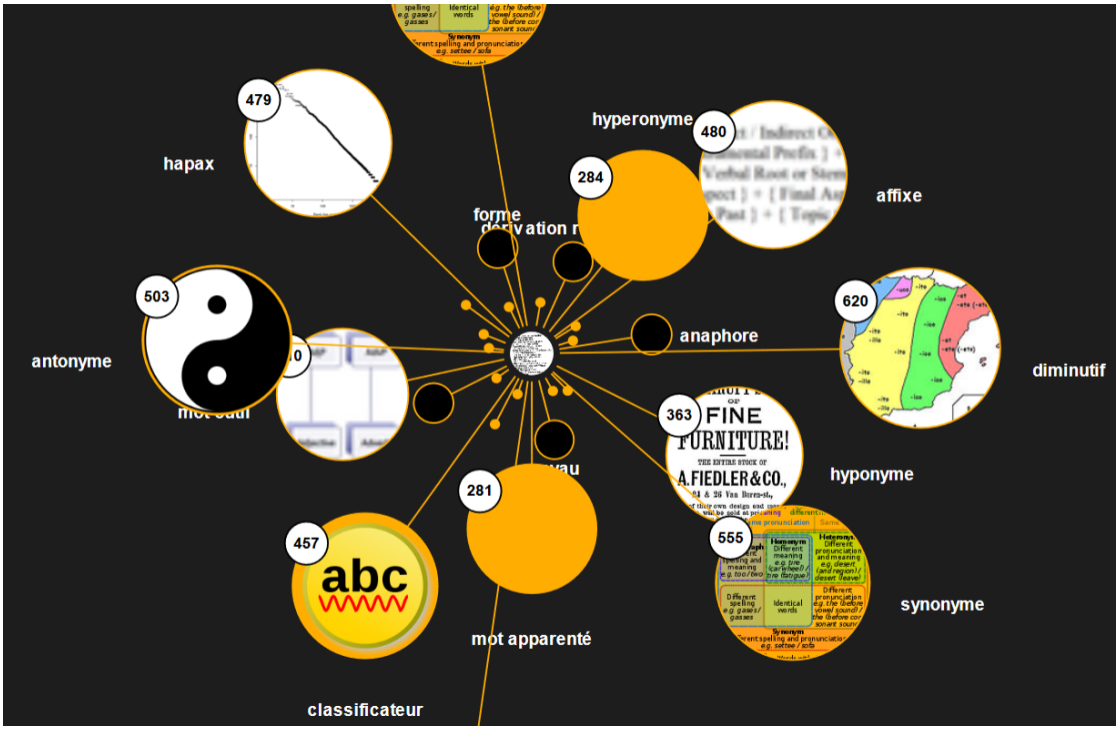


Figure 3.3: Illustration of the BabelNet semantic network of the french word "mot".

Keeping a list of word-examples associated with ids is the chosen solution because of its simplicity.

The call to disambiguation is long, and the texts requiring multiple queries add to the time constraints of the project (4.1).

3.4 Processing

The processing is used to apply existing data analysis methods (CREA Boissier (2022) using, in particular, Formal Concept Analysis Belohlavek (2008); Wille (1982) and Clustering Everitt et al. (2011), or LDA Blei et al. (2003)).

3.4.1 Filtering and calculation of the number of occurrences

Input		Output
babel-token	→	occurrences matrix

Once pre-processing is complete, two operations are carried out in parallel: filtering and calculating the number of occurrences of each concept or named entity.

With regard to filtering, this involves using the consistency score calculated via babelization during pre-processing to retain only those concepts and named entities with a certain relevance in a text. This approach makes it possible to get rid of the most complex errors that could have appeared in the OCRization phase and changed the meaning of a semantic unit. These errors are mainly of two types.

- Errors on a short semantic entity (the shorter it is, the smaller the number of errors required to lose the meaning of the concept or named entity).
- Errors on a semantic entity that is difficult to access, such as a speech bubble or image (the less accessible, the more complex the OCRization and the greater the number of possible errors).

Empirically, the value of the filtering threshold is a consistency score of 0.05 Boissier (2022).

In parallel, each concept or named entity passing the filter increments a slot in the occurrence matrix, which counts the number of each concept or named entity for each document. The creation of this matrix is in fact an information selection phase, in which the position of the concept or named entity within the text is no longer preserved, unlike the equivalent text for the CREA method.

3.4.2 Normalization

Input		Output
occurrences matrix	→	frequency matrix

Once the occurence matrix has been retrieved, its values and amplitude depend on the size and number of documents in the corpus studied, as well as the number of concepts and named entities after the filtering step.

To correct the situation and normalize the occurrence matrices, their row values are divided by the total row sum. Column normalizations are performed using the same procedure, preceded and followed by matrix transposition.

At the end of normalization, the matrix contains frequencies Jaffal (2019) between 0 and 1.

3.4.3 Formal context creation

Input		Output
frequency matrix	→	formal context

The aim of this step is to transform the frequency matrix into a Boolean matrix, named "formal context". This matrix indicates whether there is a significant link between each concept, each named entity, and each document in relation to the others.

The notion of "significant link" is intrinsic to the strategy chosen to create the formal context.

- The direct strategy
  - will act as an entry block in the chain leading to the creation of a mutual impact graph.
  - is not parameterized.
  - separates frequencies into two groups: null frequencies (set to false) and non-null frequencies (set to true).
- The high strategy
  - will serve as an input block in the chain leading to the creation of a conceptual similarity graph.
  - is parameterized by β, half the amplitude of the medium frequencies.
  - separates frequencies into three groups: low frequencies (< 0.5 - β) (set to false), medium frequencies ($\geq$ 0.5 - β and $\leq$ 0.5 + β) (set to false), high frequencies (> 0.5 + β) (set to true).

3.4.4 Analysis of the formal context

The aim of this step is to create the Galois lattice from the formal context and calculate the CREA method's metrics of mutual impact and conceptual similarity.

Input		Output
formal context	→	Galois lattice and CREA measures

The vast majority of Python libraries dedicated to Formal Concept Analysis allow the creation of the Galois lattice. For this project, the concepts library and its lattice method were chosen.

The Galois lattice is a graph that takes two types of information as input and whose nodes, called "formal concepts" contain a combination of these two types. In the chain leading to CREA visualizations, the input types are documents and concepts or named entities. Each node provides information on the content of the formal context. For example, the presence of a node containing document 1, document 2, concept 1 and concept 2 indicates that these two documents have a significant link with these two concepts. In this way, the strength of the link between one document and another can be determined by the number of significant concepts shared by these two documents. It's this kind of measurement that leads to the results of the CREA method.

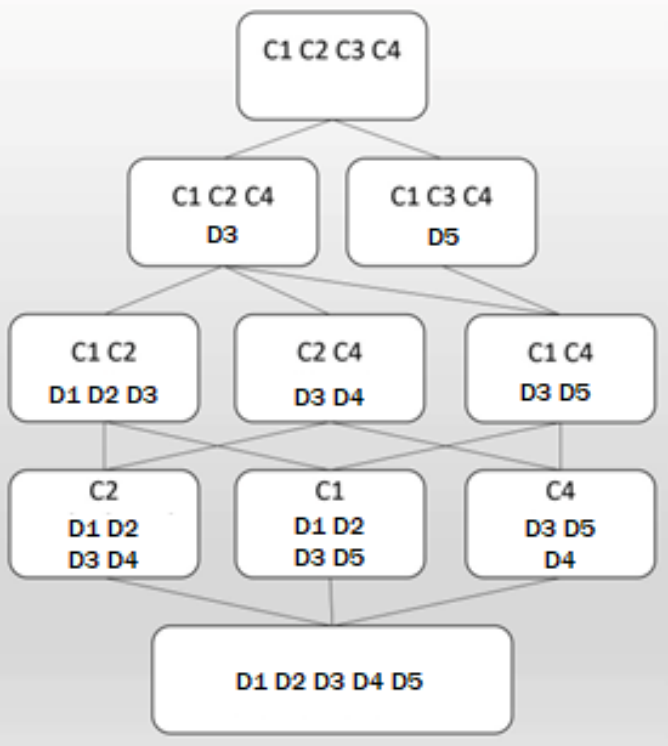


Figure 3.4: Illustration of a Galois lattice.

- Mutual impact
  - is a matrix of documents and concepts or named entities, corresponding to the ratio of the number of nodes containing both the document and the concept or named entity to the number of nodes containing one, the other or both.
  - is based on the Galois lattice calculated from the direct strategy.
- Conceptual similarity
  - is a square matrix of concepts or named entities, corresponding to the ratio of the number of nodes containing both the first and second concept or named entity to the number of nodes containing one, the other or both. By definition, all values on the diagonal of the matrix are 1, because "A and A" = "A or A".
  - is based on the Galois lattice calculated from the high strategy.

3.5 Visualization

Visualization will be used to visualize output data to highlight document vocabulary (notably using NetworkX Hagberg et al. (2008)).

Input		Output
mutual impact	→	mutual impact graph
conceptual similarity	→	conceptual similarity graph

The aim of this step is to generate a visual presentation of the metrics calculated in the previous steps. In particular, the generation of a visual presentation of these metrics that does not require the use of an external application such as Gephi Bastian et al. (2009), but runs directly within the Python code thanks to a library to be determined. Igraph Csardi et al. (2006) and Obsidian could be used for this purpose, but for this project NetworkX Hagberg et al. (2008) has been selected for its flexibility. It can be used to apply clustering algorithms, to position nodes on the plane via attraction/repulsion forces, and to display the result with various options such as node color and size, and similarly for edges.

The mutual impact graph is a bipartite graph, i.e. a graph composed of two groups of nodes, with all edges connecting a node in the first group to a node in the second. In the case of mutual impact, the groups of the bipartite graph are documents and concepts or named entities. The weighting of the edges is defined by the mutual impact values contained in the previously calculated matrix. For each time slice, this graph is used to determine the major topics covered by each document. In the context of our work, this will be the editorial line of the documents studied. By combining the different time slices, the graph will make it possible to determine the evolution of the editorial lines of the documents over time and the different historical events.

The following graphical choices have been made:

- The size of the nodes is independent of the database, to avoid aberrations. The original idea was to link size to the number of edges in the node, but this resulted in giant or tiny nodes.
- Nodes representing documents are large and blue, containing the path to the document in question.
- Nodes representing concepts or named entities are small and their color is on a gradient from red to green depending on the number of nodes where the latter is present (the more the node is present, the more its color tends towards green). The algorithm generating the various possible shades of the gradient has been created in advance, based on the number of documents in the database studied.
- The thickness of the edges represents the strength of the metric studied.
- Node placement in space depends on the Fruchterman-Reingold force-directed placement algorithm Fruchterman and Reingold (1991), which applies:
  - a repulsion force between nodes, parametrized by their distance from each other.
  - an attractive force parametrized by the strength of the edges.
  - a gravity-like force of attraction to keep nodes in the display area.

The conceptual similarity graph is a graph whose nodes are the concepts or named entities. The weighting of the edges is defined by the conceptual similarity values contained in the previously calculated matrix.

The following graphical choices have been made:

- The size of the nodes is independent of the database, to avoid aberrations. The original idea was to link size to the number of edges in the node, but this resulted in giant or tiny nodes.
- Nodes represent concepts or named entities, and their color is determined by the cluster to which they belong via the Clauset-Newman-Moore Jiang et al. (2014) greedy modularity maximization algorithm.
- The thickness of the edges represents the strength of the metric under study.
- Node placement in space depends on the Fruchterman-Reingold force-directed placement algorithm, which applies:
  - a repulsion force between nodes, parametrized by their distance from each other.
  - an attractive force parametrized by the strength of the edges.
  - a gravity-like force of attraction to keep nodes in the display area.

Conceptual similarity graphs have not yet been generated.

Chapter 4

Measures

4.1 Pre-Processing

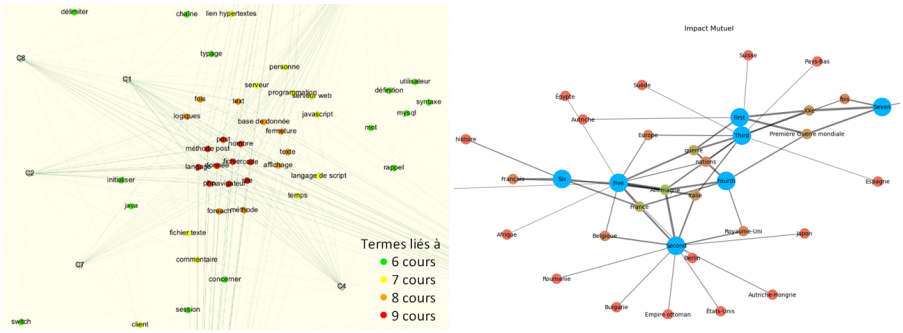
	Snowball	TreeTagger	Equivalent text
Speed	+++	+	-
Quality	-	+	+++

Snowball lemmatization is very fast and efficient on simple words, but tends to over-lemmatize. Because of its non-contextual operation, words like "tapis" ("un tapis" / "tapis dans l'ombre") will always be considered as verbs and returned to the root. Similarly, for words like "cactus" whose final s is not linked to the plural, the algorithm will consider that a final s is a suffix to be removed independently of cases. Removing endings doesn't always work for verbs, for example "vouloir" $\rightarrow$ "voul" but "voudrait" $\rightarrow$ "voudr".

TreeTagger lemmatization is fast and highly efficient. Its non-contextual operation is counterbalanced by a very detailed knowledge of the language, so it doesn't fall into the same traps as Snowball (e.g. Tapis). The return to the indicative for verbs avoids lemma modifications (e.g. Vouloir) and the return to the masculine singular for nouns avoids problems with s already present in the singular (e.g. Cactus).

The use of equivalent text as an analogy for lemmatization is slow, but extremely effective. Its context-sensitive operation, while considerably slowing down the process, enables a much finer analysis of the text. Using Babelfy's API, on the other hand, means limiting the size of queries, and therefore carrying out several processing operations, which slows down the process even more. Another disadvantage is that the number of Babelfy queries per person per day is limited (to a basic 5000, which can be increased for research work). On the other hand, the use of a semantic network (BabelNet) enables complex management of cases of compound semantic units that other methods do not (e.g. "base de données", "fruit de mer").

4.2 Visualization



(a) F. Boissier's visualizations Boissier (2022) using ForceAtlas and Gephi Bastian et al. (2008).

(b) Results obtained using NetworkX Hagberg et al. (2008) and the Fruchterman-Reingold force-directed algorithm Fruchterman and Reingold (1991).

F. Boisier's visualization deals with the use of Gephi following the application of the CREA method on a corpus made up of database courses given in higher education in computer science. This visualization uses Gephi, a tool external to the tool's pre-established code.

The comparative visualization deals with the use of NetworkX following the application of the CREA method to a corpus of texts relating to the First World War. This visualization is internal to the CREA method application tool and requires no external tool.

The result is similar.

- The documents are in the middle of the concepts and named entities, making it possible to see which ones they are close to.
- Concepts central to the text are placed at the heart of the graph, and concepts found in some or all documents are on the periphery.

Chapter 5

Discussion and Conclusion

5.1 Discussion

5.1.1 Related Work

The work carried out within this project is mainly based on Marie Puren's work in History Puren (2013, 2016) and Fabrice Boissier's thesis on the CREA method Boissier (2022). It is in fact an alteration of the method to adapt it to the processing of masses of data in History. It is also on this thesis that the comparisons and the majority of the measurements carried out are based.

5.1.2 Future Work

Perspectives for this work include applying the various tools to the data corpus, creating the conceptual similarity graph, and adding alternative methods, and libraries for processing (Word2Vec Mikolov et al. (2013), Top2Vec Angelov (2020), ETM Bagheri et al. (2020)) and visualization (Igraph Csardi et al. (2006), Obsidian).

5.2 Conclusion

This report described the application of automatic language processing techniques to the militarization of children's literature from 1860 to 1919. The study recognizes the immense potential of digitized documents and the use of computational methods for historical research. By exploiting Gallica's vast collection of digitized and OCRized texts, the project offered insight into the linguistic aspects of the militarization of children's literature during a critical period in History, while enriching the interdisciplinary approach through collaboration between computer scientists and historical researchers, offering a unique perspective on the research question.

The study also led to the development of language analysis tools that can be generalized to other research questions, paving the way for other joint studies in History and Computer Science, whether they concern other text corpora, other lexical fields, or even other historical periods.

Chapter 6

Bibliography

Angelov, D. (2020). Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*. (page 20)

Audouin-Rouzeau, S. (1993). *La Guerre Des Enfants : 1914-1918*. Armand Colin, Paris. (page 4)

Bagheri, A., Sammani, A., van der Heijden, P. G., Asselbergs, F. W., and Oberski, D. L. (2020). Etm: Enrichment by topic modeling for automated clinical sentence classification to detect patients' disease history. *Journal of Intelligent Information Systems*, 55:329–349. (page 20)

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media*. (pages 7, 16, and 19)

Belohlavek, R. (2008). Introduction to formal concept analysis. *Palacky University, Department of Computer Science, Olomouc*, 47. (pages 7 and 13)

Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72. (page 6)

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022. (pages 6 and 13)

Boissier, F. (2022). *CREA: méthode d'analyse, d'adaptation et de réutilisation des processus à forte intensité de connaissance - cas d'utilisation dans l'enseignement supérieur en informatique*. PhD thesis, Paris 1. (pages 6, 7, 13, 19, and 20)

Brosset, L. (2016). Google Livres et la numérisation : quels impacts pour les bibliothèques numériques ? Master's thesis, Université Grenoble Alpes. (page 4)

Csardi, G., Nepusz, T., et al. (2006). The igraph software package for complex network research. *InterJournal, complex systems*, 1695(5):1–9. (pages 16 and 20)

Ekinci, E. and İlhan Omurca, S. (2020). Concept-lda: Incorporating babelfy into lda for aspect extraction. *Journal of Information Science*, 46(3):406–418. (page 6)

Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster analysis*. Wiley, 5th edition. (pages 6 and 13)

Fruchterman, T. M. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164. (pages 7, 16, and 19)

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States). (pages 7, 16, and 19)

Jaffal, A. (2019). *Aide à l'utilisation et à l'exploitation de l'analyse de concepts formels pour des non-spécialistes de l'analyse des données*. PhD thesis, Université Panthéon-Sorbonne-Paris I. (pages 7 and 13)

Jiang, Y., Jia, C., and Yu, J. (2014). An efficient community detection algorithm using greedy surprise maximization. *Journal of Physics A: Mathematical and Theoretical*, 47(16):165101. (page 17)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26. (page 20)

Moro, A., Raganato, A., and Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244. (pages 6 and 10)

Porter, M. F. (2001). Snowball: A language for stemming algorithms. (page 6)

Puren, M. (2013). *Les Livres Roses de la guerre (1915-1919) : la mise en scène de l'enfant-héros pendant la Première Guerre mondiale*. Bibliothèque nationale de France, Paris. (pages 4 and 20)

Puren, M. (2016). *Jean de La Hire. Biographie intellectuelle et politique, 1870-1956*. PhD thesis, Ecole nationale des chartes. (pages 4 and 20)

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *New methods in language processing*, page 154. (pages 6 and 10)

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50. (pages 6 and 10)

Wille, R. (1982). Restructuring lattice theory: An approach based on hierarchies of concepts. In Rival, I., editor, *Ordered Sets*, volume 83 of *NATO Advanced Study Institutes Series*, pages 445–470. Springer Netherlands. (pages 7 and 13)

Thank you to those who reviewed and corrected this report prior to submission. Their valuable contributions have improved the quality and clarity of this research.