Characterizing narrative entities with embeddings : a case study of character relationships in Les Misérables

Guillaume Guex[1]

[1]Departement of Language and Information Sciences, University of Lausanne, Switzerland

Abstract

A clear and well-documented LaTeX document is presented as an article formatted for publication by CEUR-WS in a conference proceedings. Based on the "ceurart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

Keywords

LaTeX class, paper template, paper formatting, CEUR-WS

1. Introduction

In the field of *Digital Humanities*, *Distant Reading* tools (REF) allow researchers to quickly gain knowledge on textual corpora without actually reading them. Purposes of these methods are various, but can be mainly categorized into two groups: in the first case, these methods are used in order to tag, classify, or summary large quantities of texts, in order to quickly structure information or to deliver a speech over the whole studied corpus (REF). Methods in this case rely heavily on Big Data and make an extensive use of Machine Learning, often with the help of supervised methods. In the second case, researchers use computational methods to underline hidden structures in a small corpus or even a lone text, which helps them to refine their understanding of this corpora and reinforce stated hypotheses (REF). Methods in this setting can also rely on Machine Learning, but must typically be build with more caution and attention to details: corpora are small, analyses are closer to the work, and methods must be more transparent in order to appropriately interpret results. The use of exploratory tools and unsupervised methods is also preferred in this context, as it is less desirable to base methods on information coming from large external corpora. The proposed method typically belong to this second group of methods, as it is unsupervised and work on a single document.

When a lone (or a few) textual narrative is analyzed, a particular effort is made into studying *narrative entities* (characters, event, location, etc.) used by the author in her/his book. Researchers are often interested by characterizing them and see how they are articulated to one another in the story. Various computational tools can help them in this tasks, to name a few, Named Entity Recognition tools (REF), Automatic Character Networks Extraction (REF), Sentiment Analysis (REF), Topic Modeling (REF), Textometry (REF), and Word Embeddings (REF). All these methods have been used in order to explicitly show hidden structures constructed by the author in her/his work. It permits to find regularities and patterns, and can help to categorize particular narrative constructions, writing styles, or genres. These kind of methods can be great complement to classical analyses of literacy works as they allow to efficiently summarized information which is otherwise quite diffuse.

In this article, we propose a general framework in order to automatically characterize various narrative entities in a literacy work. The entire framework is exposed starting from a very wide perspective, which is how to organize the textual data, and is narrowed down to a very specific use, the study of character relationships in *Les Misérables*, by Victor Hugo. Along this presentation, various choices are made to highlight a particular usage of this framework, but theses choices should be viewed as suggestion rather than rules: the real strength of this framework is its flexibility and the direction taken in this article is oriented for a particular task. To be more specific, we are interested here in using *embeddings* in order to locate *character relationships* alongside the vocabulary. Similarity measures can then be constructed between these words and the relationships, which can help a practitioner to characterize them. Four variations of this method are proposed, and are tested on *Les Misérables*.

The idea behind this methodology comes from the field of automatic extraction and analysis of *character networks* from literacy works (REF). When building character networks from a textual narrative, the most widespread method consists in dividing the studied work into n *textual units* $u_1, \dots, u_n$, which can be, e.g., sentences, paragraphs, or chapters, and then counting characters co-occurrences in these units (REF). Usually, the

COMHUM 2022: Workshop on Computational Methods in the Humanities, June 9–10, 2022, Lausanne, Switzerland

guillaume.guex@unil.ch (G. Guex)

0000-0003-1001-9525 (G. Guex)

text constituting these units is discarded and the resulting network displays edges which roughly represent an aggregated number of interactions between characters. However, by doing so, the aggregation occurs on various type of interactions and will give little information about the type of relationship which exist between characters. Various improvements where proposed in order to weight or sign (or both) the edges in the character networks, such as using Sentiment Analysis (REF) Topic Modeling (REF), or BLABLA (REF). A particular inspiration for the current work is the article by (REF), where authors also analyzed characters in *Les Misérables* by building various signed and weighted networks, with the help of Sentiment Analysis and Topic Modeling. The current framework was build by expanding this idea of refining character relationships while keeping exploring directions as wide as possible. Embeddings appeared to us to be the proper tool for achieving this. As a matter of fact, with embeddings, the textual contents of units are transformed into workable mathematical objects (the vectors), usable for various tasks, while conserving a maximum of information. The framework has been further generalized in order to be applicable on various type of narrative entities, but the presented case remains the study of character relationships in *Les Misérables*.

The current article is structured as followed. Section 2 define the methodology, with section 2.1 defining the data organization, section 2.2 describing how to embed textual units, and section 2.3 deriving entity vectors lying in the same space as units. In section 3, we present the methods and results for the case study of character relationship embeddings in *Les Misérables*, and section 4 draw conclusions and perspectives about this work. All (Python) scripts and datasets used in this article, as well as extended results, can be found in the dedicated GitHub repository.[1]

2. Methodology

2.1. Data organization

In this article, the textual narrative is divided in n *textual units* $u_1, \dots, u_n$, and is represented through two tables. The first one is well known in the field of textual analysis, and consists in the $(n \times v)$ *unit-word contingency table* $\mathbf{N}$, as represented by Table 1, where v is the vocabulary size. In this table, each row represents an unit, each column a word, and cells n_{ij} counts the number of times the word i appears in the unit. Using this table typically denotes a *Bag-of-Words* approach in our analyses.

The second table is the *unit-entity table*, noted $\mathbf{E}$. It has a size of $(n \times p)$ where p is the number of *narrative entities* found in the text and cells e_{ij} indicates the presence, or the count for a weighted version, of entity j in unit i. An narrative entity, in the context of this article, can be loosely defined in order to be flexible for various types of works or analyses. It can roughly be seen as a recurring narrative object, which can contain one or more characters. For example, it can be a location, a character, a pair of characters (or even a triplet, a quadruplet, etc.), an oriented character interaction (e.g. a dialog), or even a particular recurring event containing multiple characters (e.g. a meeting). In this article, we mostly consider character and pair of characters as entities, as shown in Table 2. Note that we consider that a character or a pair of characters are present in the unit if character names (or aliases) are detected above a fixed threshold. A weighted version of this table, where e_{ij} contain the number of occurrences of the entity j in the unit i, is also possible and formulas are written accordingly.

	aller	allumer	apercevoir	bas	bon	...
u_{101}	23	2	6	11	6	...
u_{102}	12	1	0	3	9	...
u_{103}	10	0	5	1	5	...
u_{104}	0	0	1	0	0	...

Table 1
A snippet of the contingency table $\mathbf{N}$ extracted from *les Misérables*. Rows are chapters, columns are words in the vocabulary, and cell n_{ij} counts the number of time word j appear in chapter i.

	Cosette	Thénardier	Valjean	...
u_{101}	1	1	0	...
u_{102}	1	1	0	...
u_{103}	1	1	0	...
u_{104}	1	0	1	...

	...	Cosette-Thénardier	Cosette-Valjean	...
u_{101}	...	1	0	...
u_{102}	...	1	0	...
u_{103}	...	1	0	...
u_{104}	...	0	1	...

Table 2
A snippet of the entities table $\mathbf{E}$ extracted from *les Misérables*. Rows are chapters, columns are characters (top) and character-pairs (bottom), and cell e_{ij} denotes if j appear in chapter i.

This data organization already gives an orientation to the subsequent analyses and should be kept in mind of the practitioner. Textual unit are now considered as *individuals* (in the statistical terminology), defined by their *variables* contained in the different columns of both tables. Moreover, subsequent analyses are oriented in searching how the entity table $\mathbf{E}$ has an influence over the contingency table $\mathbf{N}$, i.e. searching which words are over-represented or under-represented knowing entities in the unit. While authors use characters in order to build the narrative, we, to a certain extent, work backward: we are searching how character appearances and interactions

[1] https://github.com/gguex/char2char_vectors.

in the textual unit act on her/his choice of words. If the extraction method permits it, a practitioner should include all entities which she/he desire to study. Here, for example, the choice to include character-pairs along with characters is motivated by the fact that we are interested in studying character relationships. A character-pair can roughly be seen as an interaction between two characters, and this interaction should be considered as an object of its own: the presence of this interaction in a unit do not result in having a mixture of words used for each character, but rather gives a specific flavor to the unit.

This data organization also highlights the importance of choosing a proper size for the units. These units should be large enough to contain enough words in order the properly capture the textual specificity of each unit, but not too large, as each unit should ideally captures particularities about one of the entity. Unfortunately, it is impossible to define an ideal size for all types of analysis and this size should be balanced regarding the level of analysis, the text size, the selected entities, and previous knowledge of the studied work.

The use of a contingency table $\mathbf{N}$ to represent the textual resource present in the units denotes a *Bag-of-Words* approach. Using this approach loses the information relative to the order of words in the units, but permits to transform a chain of characters, improper to statistical analyses, into a contingency table, a well studied mathematical object which allows the use of various kind of statistical methods. The next section shows a particular direction on how to use this table, with the help of *embeddings*.

2.2. Embedding of textual units

Various methods can be performed on the contingency table $\mathbf{N}$ in order to extract information from it, such as Topic Modeling (REF), Sentiment Analysis (REF), or Textometry (REF) methods. Here, we make to choice to extract a lower dimensional, numeric representation of each units, in other words, an *embedding of textual units*. In section 2.3, these vectors of textual units are used as anchor points in order to also embed entities into the same space. Therefore, it is crucial that an interpretation about the directions or the regions of this embedding is possible, in order to properly interpret the localization of entity vectors (the relative position of entity vectors among themselves is generally insufficient). For that reason, we focus on embeddings of textual units which also contain *vectors of words*: by examining the positions of entities relatively to word vectors, entities can be characterized. We propose two embeddings verifying this condition: Section 2.2.1 describes *Correspondence Analysis (**CA**)* and section 2.2.2 focuses on *Pre-trained Word Vectors (**WV**)*.

2.2.1. Correspondence Analysis (CA)

Using *Correspondence Analysis* (**CA**) in order to analyze textual resources has a long tradition (REF). It has the advantage to naturally provide an embedding space, the factorial map, where units are placed alongside word vectors, and allows the interpretation of the placement of units in terms of word frequency profiles. Units and words vectors in the embedding space have a direct interpretation in terms of chi2 distance between profiles.

By performing a Correspondence Analysis on table $\mathbf{N}$, we get n vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ corresponding to units (rows) and v vectors $\mathbf{w}_1, \dots, \mathbf{w}_v$ corresponding to words (columns). Each of these vectors has a size of $\min(n, v) - 1$, which will generally be $n - 1$. For a detailed computation of quantities in CA, see Appendix A.1.

The association between a particular unit i and a word j is expressed through the scalar product between their vectors $s_{ij} := \mathbf{x}_i^\top \mathbf{w}_j$, which act as a similarity measure. A positive (resp. negative) similarity denotes a over-representation (resp. under-representation) of the word j in i, which permit to find lists of words characterizing the different units. Observe that in this article, this similarity is rather computed between a word vector and a entity vector, since the latter, as we will see in section 2.3, lies in the same space as unit vectors.

Note that vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ obtained from **CA** reflect textual unit profile (in terms of words) regarding the mean profile (the origin in the factorial map). This analysis is thus *contrastive*: it highlights unit variations among the studied text. It means that the particular tone of the whole studied text might be hidden in this analysis and only the variation around this tone will be revealed. It might lead to the situation where the (absolute) feeling experienced by the reader will not appear in this analysis, e.g. a sad character in a sad book will appear joyful if he is less sad than the mean tone. This could becomes problematic when this method is used sequentially to study multiple works: particularities of each book will be hidden. Another limitation with this approach is that the words helping the interpretation of units (and entities) are contained in the studied text. Approaches requiring to study the position of units and entities relatively to a predefined list of words (e.g., friends, enemies, family) might therefore be impossible if these words do not appear in the text.

2.2.2. Pre-trained Word Vectors (WV)

Pre-trained Word Vectors (**WV**), based on methods such as Word2Vec (REF), Glove (REF), or fastText (REF), have recently received great attention from various fields (REFS). They are generally obtained through a training on a very large corpora, such as Wikipedia (REF) or Common Crawl (REF), and the resulting embedding contains a large quan-

tity of word vectors. As shown by multiple studies (REF), these vectors are placed in order to reflect semantic and syntactic similarities between words, and are used in various applications (REF).

There exists multiple methods in order to find vectors for a *group of words*, such as sentences (REF), paragraphs (REF), or documents (REF), based on pre-trained word vectors. These derived vectors are often used to apply a classification or clustering algorithm (REF) on the newly embedded objects, or to query information (REF). In order to derive vectors for groups of words, the majority of methods use the frequencies of words in the objects, i.e. a table similar to $\mathbf{N}$, but apply various weighting scheme and normalization in order to reduce the effects of frequent words and to standardize vectors. In the present article, we use a recent methodology proposed in (REF) as it is compatible with multiple unit sizes and gives state-of-the-art results in many tasks.

Textual units vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ are obtained through the table $\mathbf{N}$ and with the method detailed in Appendix A.2. The main way to interpret units with this method is to use the cosine similarity between its corresponding vector $\mathbf{x}_i$ and every word-vectors $\mathbf{w}_j$, as defined by $s_{ij} := \frac{\mathbf{x}_i^\top \mathbf{w}_j}{\sqrt{\mathbf{x}_i^\top \mathbf{x}_i \mathbf{w}_j^\top \mathbf{w}_j}}$. The cosine similarity also permits to compare units between themselves.

With the pre-trained word vector method, the unit vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ (and entity vectors in section 2.3) lie in an *absolute space* defined by the pre-trained word vectors. Comparison between different texts are therefore more pertinent in this space and comparison with words absent from the corpus can be made. However, it is still possible that all units from a given text will be located in the same region of the space, if the style of the text is particular. In this case, the list of most associated word-vectors might be similar for every unit, and the analysis will not give satisfying results. This effect is fortunately limited by the centration of unit-vectors which occurs in the method describe in Appendix A.2.

2.3. Entity embeddings

The main goal of this article is not to analyze units, but rather the entities, i.e. the p columns of table $\mathbf{E}$. While we used the table $\mathbf{N}$ to build our embeddings of units, we now use the table $\mathbf{E}$ in order to build the entity vectors $\mathbf{y}_1, \dots, \mathbf{y}_p$ relatively to unit vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Two propositions of method are made: the *centroids method* (**CENT**) is describe in section 2.3.1 and the *regressions method* (**REG**) is explained in section 2.3.2. Both methods work with the aforementioned embeddings of units.

2.3.1. Centroids (CENT)

This method is the most trivial and is based on the following intuition: an entity is characterized by all units in which it appears. In other words, we can define the vector $\mathbf{y}_k$ for character-object object k as

$$\mathbf{y}_k = \sum_{i=1}^{n} \frac{e_{ik}}{e_{\bullet k}} \mathbf{x}_i \tag{1}$$

i.e. the center of mass, or *centroid*, of the units containing the entity, weighted by the relative weight $\frac{e_{ik}}{e_{\bullet k}}$. This way of building entity vectors is closely related to the treatment of *supplementary variables* found in **CA**: these variables do not act in the choice of factorial axis, but can still be represented afterward. However, by contrast, entity vectors are not dilated after computing centroids, which means that they lie in the same space as units (row).
An important remark about the centroid method, is that entity vectors positions are *additive*, which means that we have

$$e_{ik} = \sum_{g \in \mathcal{G}} e_{ig}, \forall i \implies \mathbf{y}_k = \sum_{g \in \mathcal{G}} \frac{e_{\bullet g}}{e_{\bullet k}} \mathbf{y}_g, \tag{2}$$

where $\mathcal{G}$ is a subset of entities. This property can be interpreted as followed: if a character k can be divided among different situations g (the character alone, the character in interaction with another character, etc.), the character vector $\mathbf{y}_k$ is in fact the centroid of all vectors $\mathbf{y}_g$ of these situations. This is not necessary an undesirable property, but it implies that the specificities of the lone character might be hidden if he is often registered in an interaction. By contrast, if we consider that an interaction between two characters is an *emerging situation*, unrelated to prior behaviors of characters, the regressions method described in the next section seems more appropriate.

2.3.2. Regressions (REG)

When building a regression model with multiple explanatory variables, it is possible to include every variables and their *interactions*. By doing so, we suppose that the effect of raising both variables is not the same as raising each variable independently. Regression models seem therefore appropriate to capture specificities of having a particular entity in a textual unit. For example, in the case of character pairs, the presence of a character a will have a effect on the vocabulary of an unit, the presence of another character b will have another effect, and the presence of the pair $\{a, b\}$ yet a different effect. Now, dependent variables in regression models still need to be defined. In fact, we are doing d regressions, with d the number of dimensions of the embeddings, and each

regression is constructed to predict the α-th coordinate of each units by using variables in the table $\mathbf{E}$. In a matrix notation, all models can be written as

$$\mathbf{X} = \widetilde{\mathbf{E}}\mathbf{B} + \mathbf{\Sigma}, \tag{3}$$

where $\mathbf{X} = (x_{i\alpha})$ is the $(n \times d)$ matrix containing unit-vectors (on rows), $\widetilde{\mathbf{E}}$ is the matrix $\mathbf{E}$ with an first additional column of 1 for the intercept, $\mathbf{B} = (\beta_{k\alpha})$ is the $((p+1) \times d)$ matrix containing intercepts and regression coefficients (each regression is contained on a column), and $\mathbf{\Sigma}$ the $(n \times \alpha)$ matrix containing normal errors.

Estimates $\widehat{\mathbf{B}} = (\widehat{\beta}_{k\alpha})$ for the intercept and coefficient regressions are in fact the researched embeddings for entities as well as for the intercept representing the general tone of the studied text. We therefore denote these estimates with $\mathbf{Y} = (y_{k\alpha})$ in the following, with the convention $y_{0\alpha}$ for intercept coefficients.

As the number of entities (i.e. predictors) might be very large, it is a good idea to add a L^2 regularization term in the objective function. Moreover, the quadratic error rate should also be weighted by the number of tokens in each unit. Including all this, we find the solution for our character-based vectors $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_p$ contained in rows of $\mathbf{Y}$ with

$$\mathbf{Y} = (\widetilde{\mathbf{E}}^\top \mathbf{Diag}(\mathbf{f})\widetilde{\mathbf{E}} + \lambda \mathbf{I}_{(p+1)})^{-1}\widetilde{\mathbf{E}}^\top \mathbf{Diag}(\mathbf{f})\mathbf{X}, \tag{4}$$

where $\mathbf{Diag}(\mathbf{f})$ is the diagonal matrix containing weights of units $\mathbf{f} = \left(\frac{n_{i\bullet}}{n_{\bullet\bullet}}\right)$, $\lambda > 1$ is the regularization coefficient, and $\mathbf{I}_{(p+1)}$ is the identity matrix of size $((p+1) \times (p+1))$.

An interesting effect of the regularization coefficient is that if λ is high, equation (4) becomes $\mathbf{Y} \approx \frac{1}{\lambda}\widetilde{\mathbf{E}}^\top \mathbf{Diag}(\mathbf{f})\mathbf{X}$, which is similar to equation (3) (with a contraction and a different weighting scheme). In fact, the regressions method with a regularized term interpolate between the hypothesis where we suppose that every entity should be considered independently (with $\lambda \to 0$), to the hypothesis of additive mixture between entities (with $\lambda \to \infty$), as discussed in section 2.3.1. Choosing an appropriate λ according to the study (how is another, difficult question) might lead to a situation revealing desirable information about entities.

3. Case studie : *Les Misérables*

For the moment, it is not possible to evaluate the presented methods by the use of some measurements, which would allow to test its validity on various corpora. In order to see if the methods give coherent results, we have to carefully scrutinized and compared them with previous knowledge of the studied work. For this reason, and because of method variations and multiplicity of the results (and lack of place), we chose to present here only one case studies: *Les Misérables* by Victor Hugo. The choice of this work is motivated by the fact that it is a large corpus, well-known, immensely studied, and containing various colorful characters and characters relationships. It is therefore a solid choice to clearly illustrate the potential of the presented methodology.

3.1. Preprocessing

The five volumes of *Les Misérables*, in French, were extracted from *Project Gutenberg*[2], while headers and footers of each files were manually removed. The whole text was lower cased, lemmatized, and stopwords[3] and punctuation were removed. Volumes, books, and chapters breaking points were kept for later uses.

We chose to use chapter as textual units. The table $\mathbf{N}$ (Figure 2.1) was build by considering words appearing at least 20 times in the text and resulted in a table of size 365 chapters $\times$ 1974 words.

Characters were detected using *Flair*[4] (REF) NER tools. In order to unify character and to further refine the detected characters list, we used hand-made lists of character and aliases from NER results. It resulted in the detection of 54 characters. The entities considered in table $\mathbf{E}$ (Figure 2.1) are single character (54 entities) and character pairs (547 entities), resulting in a table of size 365×601. A character or a character-pairs presence is considered present iff characters are detected at least 2 times in the chapter.

Note that, in section 3.3.3, we also tested experiments with entities consisting in character and character-pairs as found in each volumes (e.g. Cosette-Valjean in volume one and Cosette-Valjean in volume two are now two different objects), with the addition of volume constants ($V_i = 1$ in volume i and $V_i = 0$ in other volumes) in order to isolate specific volume vocabulary. This new table $\mathbf{E}_{\text{vol}}$, containing 1124 entities, permits to see a diachronic evolution of words describing volumes, characters, and character relationships.

3.2. Methods

There are two types of methods for unit embeddings, **CA** (section 2.2.1) and **WV** (section 2.2.2), as well as two methods for entity embeddings, **CENT** (section 2.3.1) and **REG** (section 2.3.2), making a total of 4 possibles ways for obtaining final embeddings.

The **CA** method is fully automated, and results in vectors in a 364-dimensional space, while the **WV** methods is based on pre-trained word vectors using *fastText* (REF)

[2] https://www.gutenberg.org/.
[3] from a list made by Jacques Savoy http://members.unine.ch/jacques.savoy/clef/frenchST.txt.
[4] https://github.com/flairNLP/flair.

trained on Common Crawl.[5] For French, the number of word-vectors m is 2 millions and the dimension of the vector space is 300.

Note that, in addition to having two table 4 variations for the methods, 2 tables $\mathbf{E}$ and $\mathbf{E}_{\text{vol}}$, and a considerable number of entities, results can also be presented in various ways (similarities between entities, between entities and words, etc.). Thus, we chose to show here a selection of results for the each method: the 5 most associated words regarding entities (section 3.3.1), the 5 most associated entities regarding a subset of words (section 3.3.2), and a diachronic study of top associated words for a subset of entities (section 3.3.3). We invite curious readers to consult results for all words and entities, which can be found in our GitHub repository.[6]

3.3. Results

3.3.1. The most associated words for a subset of entities

The first results in this section is to examine the most associated words with a subset of entities, as measured by the similarities defined in section 2.2. Results can be found in Table 3 for all methods.

We can first observe that **CA** methods seems to summarize entities with a vocabulary closer to the work, while **WV** methods tend to frequently use words with a wider scope, with notably more verbs. It results in having the **WV** giving a general feeling for the tone used for describing characters and relationships, while the **CA** method can depict very specific objects, location or event associated with these entities. This behavior can understood by the nature of unit embeddings: in the **WV** embedding, word-vectors are fixed and do not consider the actual frequencies of words found in the studied corpora. A character can be close to a word appearing only a few times (or none) in the corpus if this word is located near the vocabulary associated with this character, as semantically similar words are located in the same region of space. By contrast, **CA** will generally takes into account word frequencies along with specificities in order to describe an entity, and very specific words can define particular regions of space.

Another remark can be made about the difference between **CENT** methods and **REG** methods. As expected, we see that the **CENT** methods reveal their additive construction between characters and relationships: words used to describe a relationship rob off on their characters descriptions (see e.g. Cosette, Cosette-Marius, and Cosette-Valjean). By constrast, the **REG** methods display more "perpendicular" descriptions of entities, with less words repeating.

Note that we did not show here the least associated words with each entities, as they are frequently the same for all methods and all related to the long description of the Battle of Waterloo in volume 2 ("infanterie", "wellington", "cuirassier", "bridage"), containing no protagonist of the story.

Overall, we find that the **CA-REG** method gives the most satisfying results, with pertinent words associated with each entity, and an high variety in the choice of words.

3.3.2. The most associated characters-based objects for a subset of words

These results are extracted from the transposed table from the last, and display the most associated entities to a selected set of words. They can be found in Table 4. This type of results can be seen as queries, made from a single word, which output the most associated entities in the work related to that query. We chose here to show top entities related to words "aimer", "rue", "justice" and "guerre", as they represent some of the main topics of the book. In this task again, from our point of view, the **CA-REG** displays the most accurate results: the main love relationship (Cosette-Marius) of the book is the most associated entity for "aimer", several "amis de l'ABC" (a revolutionary group) are most associated with "rue", the cop-suspect relationship (Javert-Valjean) is the top entity for "justice", and military officers or bellicose characters are associated with "guerre". While somewhat inferior with the selected set of queries, **WV** methods have the advantage to be able to query words outside the scope of the book, as the pre-trained word embedding possess a very large vocabulary.

Note that another way to represent these results is through weighted signed networks, as found in Figure 1 (for **CA-REG**). The network structure represent the number of time characters are detected together (which do not depend on the query), and the signed weights (edge color) display similarities between character relationship (edges) and the queried word. This representation gives a quick visual support in order to explore the studied work and could be implemented as a standalone program.

3.3.3. A diachronic study of the most associated words for a subset of character-based objects

These results are obtained from the table $\mathbf{E}_{\text{vol}}$ where entities are considered different based on the volume. By doing so, it permits to track the evolution of similarities along the book. Additionally to entities, we can also define constant term V_i for each volumes i, which ab-

[5] https://fasttext.cc/docs/en/crawl-vectors.html.

[6] the "results" folder in https://github.com/gguex/char2char_vectors.

	Cosette	Cosette-Marius	Cosette-Valjean	Marius	Valjean
CA-CENT	poupée (0.7)	**noce** (1.72)	**noce** (1.0)	théodule (0.61)	**mestienne** (0.51)
	noce (0.68)	**mariage** (1.31)	**mestienne** (0.97)	jondrette (0.59)	fossoyeur (0.46)
	mestienne (0.58)	**marié** (1.21)	**mariage** (0.71)	**ursule** (0.56)	**accusé** (0.45)
	mariage (0.48)	marier (1.11)	**marié** (0.68)	vernon (0.53)	maire (0.39)
	marié (0.48)	baron (1.0)	corbillard (0.65)	tante (0.52)	jean (0.37)
	Marius-Valjean	Javert	Javert-Valjean	Myriel	Myriel-Valjean
	noce (1.2)	**accusé** (1.47)	**accusé** (1.85)	conventionnel (5.03)	chandelier (6.28)
	mariage (0.85)	arras (1.04)	**avocat** (1.12)	évêque (3.54)	gendarme (5.06)
	ursule (0.85)	mouchard (0.97)	**preuve** (1.1)	oratoire (3.39)	panier (4.72)
	marié (0.8)	**avocat** (0.96)	président (1.08)	hôpital (2.57)	couvert (4.64)
	tableau (0.74)	**preuve** (0.93)	forçat (1.01)	cathédrale (2.54)	deuil (4.52)
CA-REG	Cosette	Cosette-Marius	Cosette-Valjean	Marius	Valjean
	seau (1.23)	amant (0.83)	blessure (0.78)	jondrette (1.76)	matelas (1.02)
	poupée (0.86)	mariage (0.73)	**noce** (0.76)	réchaud (1.26)	**chandelier** (0.87)
	ravissant (0.7)	entraîner (0.7)	file (0.6)	galetas (1.11)	toulon (0.82)
	source (0.65)	**noce** (0.67)	corbillard (0.58)	bouge (1.05)	fossoyeur (0.79)
	rassurer (0.61)	volupté (0.63)	mestienne (0.58)	tableau (0.93)	pelle (0.76)
	Marius-Valjean	Javert	Javert-Valjean	Myriel	Myriel-Valjean
	égout (1.1)	arras (1.09)	accusé (1.04)	conventionnel (2.99)	deuil (1.14)
	vase (1.08)	roue (0.89)	nier (0.79)	évêque (1.76)	**chandelier** (1.07)
	issue (1.07)	bonjour (0.83)	quai (0.54)	cathédrale (1.14)	aveugle (1.01)
	sable (1.0)	malle (0.8)	avocat (0.53)	prêtre (1.11)	panier (0.94)
	couloir (0.98)	cabriolet (0.76)	fonction (0.5)	philosophie (1.06)	gendarme (0.89)
WV-CENT	Cosette	Cosette-Marius	Cosette-Valjean	Marius	Valjean
	jean (0.34)	aimer (0.38)	**jean** (0.6)	embrasser (0.36)	**jean** (0.56)
	dormir (0.28)	rêver (0.34)	**jacques** (0.3)	**essayer** (0.36)	**habiller** (0.27)
	regarder (0.26)	**vouloir** (0.32)	**philippe** (0.26)	**avouer** (0.36)	**poser** (0.26)
	habiller (0.26)	douter (0.32)	**habiller** (0.26)	**vouloir** (0.35)	**jacques** (0.26)
	voir (0.25)	**avouer** (0.32)	**pantalon** (0.25)	**voir** (0.35)	**pantalon** (0.25)
	Marius-Valjean	Javert	Javert-Valjean	Myriel	Myriel-Valjean
	jean (0.35)	**saisir** (0.34)	**jean** (0.54)	**évêque** (0.59)	**évêque** (0.55)
	questionner (0.31)	**jean** (0.34)	denis (0.31)	**archevêque** (0.52)	**archevêque** (0.46)
	essayer (0.31)	placer (0.31)	**jacques** (0.3)	**prêtre** (0.45)	**prêtre** (0.42)
	oser (0.31)	retirer (0.29)	**saisir** (0.3)	**abbé** (0.39)	âme (0.42)
	poser (0.29)	dégager (0.29)	**philippe** (0.28)	souverain (0.38)	**abbé** (0.39)
WV-REG	Cosette	Cosette-Marius	Cosette-Valjean	Marius	Valjean
	contempler (0.29)	éternel (0.35)	**rue** (0.44)	regarder (0.38)	**jean** (0.56)
	emplir (0.29)	**amour** (0.35)	**jean** (0.41)	voir (0.36)	pantalon (0.28)
	doucement (0.27)	humanité (0.34)	faubourg (0.41)	refermer (0.34)	jacques (0.26)
	envelopper (0.26)	**âme** (0.32)	**boulevard** (0.41)	**glisser** (0.34)	philippe (0.23)
	illuminer (0.26)	vérité (0.32)	quartier (0.34)	poser (0.31)	**glisser** (0.23)
	Marius-Valjean	Javert	Javert-Valjean	Myriel	Myriel-Valjean
	rue (0.35)	serrer (0.34)	**rue** (0.35)	**évêque** (0.43)	ange (0.37)
	boulevard (0.35)	**glisser** (0.34)	**boulevard** (0.34)	divin (0.4)	**évêque** (0.31)
	souterrain (0.35)	forcer (0.34)	autorité (0.33)	humble (0.39)	**âme** (0.31)
	bastille (0.35)	bouger (0.33)	civil (0.33)	bonté (0.38)	**amour** (0.29)
	carrefour (0.34)	aller (0.32)	loi (0.33)	archevêque (0.37)	aurore (0.28)

Table 3
The 5 most associated words (similarity in parentheses) to a selected set of entities, regarding **CA-CENT**, **CA-REG**, **WV-CENT**, and **WV-REG** methods. Words appearing at least two times within the same method are in bold.

sorbs the associated words with each volume. Results for constants and a subset of entities (Valjean, Cosette, Cosette-Valjean) can be found in Table 5. Note that we did not show **CONT** results in this table, as they are similar to the one found in Table 3 : words are often repeated themselves for different objects and are less convincing.

Here again, we see that associated words for the **WV** give a general tone to tomes and entities, while **CA** results are more specific and related to particular events which occurred for characters. As expected, volume con-

	aimer	rue	justice	guerre
CA-CENT	Dahlia-Fameuil (1.12)	Courfeyrac-Fauchelevent (0.79)	Azelma-Babet (1.09)	Combeferre-Fauchelevent (1.35)
	Dahlia-Listolier (1.12)	Courfeyrac-Toussaint (0.79)	Azelma-Brujon (1.09)	Feuilly-Valjean (1.27)
	Fameuil-Zéphine (1.12)	Eponine-Fauchelevent (0.79)	Azelma-Claquesous (1.09)	Feuilly-Marius (1.22)
	Listolier-Zéphine (1.12)	Eponine-Gavroche (0.79)	Azelma-Magnon (1.09)	Lesgle-Valjean (1.15)
	Gillenormand-Toussaint (1.11)	Eponine-Pontmercy (0.79)	Azelma-Montparnasse (1.09)	Mabeuf-Valjean (1.15)
CA-REG	Cosette-Marius (0.35)	Courfeyrac (0.37)	Javert-Valjean (0.37)	Grantaire-Pontmercy (0.94)
	Myriel (0.31)	Grantaire-Prouvaire (0.29)	Champmathieu-Valjean (0.37)	Grantaire (0.56)
	Basque-Fauchelevent (0.25)	Cosette-Javert (0.26)	Myriel (0.21)	Marius-Pontmercy (0.55)
	Myriel-Valjean (0.22)	Marius-Prouvaire (0.22)	Grantaire (0.18)	Pontmercy (0.55)
	Fauchelevent-Gillenormand (0.21)	Enjolras (0.22)	Grantaire-Javert (0.16)	Enjolras (0.49)
WV-CENT	Cosette-Marius (0.38)	Grantaire-Prouvaire (0.57)	Azelma-Babet (0.34)	Grantaire (0.32)
	Fantine-Marius (0.34)	Marius-Prouvaire (0.54)	Azelma-Brujon (0.34)	Combeferre-Lesgle (0.32)
	Fantine-Pontmercy (0.34)	Cosette-Javert (0.43)	Azelma-Claquesous (0.34)	Feuilly-Lesgle (0.25)
	Basque-Fauchelevent (0.34)	Magnon-Monsieur Thénardier (0.37)	Azelma-Magnon (0.34)	Combeferre-Marius (0.25)
	Prouvaire-Valjean (0.33)	Gavroche (0.36)	Azelma-Montparnasse (0.34)	Combeferre-Grantaire (0.25)
WV-REG	Prouvaire-Valjean (0.34)	Grantaire-Prouvaire (0.8)	Champmathieu-Valjean (0.38)	Grantaire-Pontmercy (0.39)
	Champmathieu-Chenildieu (0.31)	Marius-Prouvaire (0.77)	Azelma-Brujon (0.35)	Enjolras-Marius (0.35)
	Brevet-Chenildieu (0.31)	Courfeyrac (0.66)	Azelma-Claquesous (0.35)	Grantaire (0.31)
	Brevet-Cochepaille (0.31)	Cosette-Javert (0.64)	Azelma-Magnon (0.35)	Combeferre-Lesgle (0.31)
	Champmathieu-Cochepaille (0.31)	Prouvaire (0.63)	Azelma-Montparnasse (0.35)	Cosette-Gavroche (0.27)

Table 4
The 5 most associated entities (similarity in parentheses) to a selected set of words, regarding **CA-CENT**, **CA-REG**, **WV-CENT**, and **WV-REG** methods.

stants give a quick summary of each volumes, especially with the **CA-REG** method (e.g. V_2 for the Battle of Waterloo, V_4 for the barricade event). Associated words with entities also seems accurate in describing them. Note that Cosette was not detected in volume 3 because she is not explicitly cited (she is often refereed as "the daugther of M. Leblanc"), and this also explains the absence of the Cosette-Valjean pair.

4. Conclusion

In this article, we introduced a general framework in order to automatically extract textual information about narrative entities from a small corpus or a single work. The framework is build on two tables, the unit-word table $\mathbf{N}$ and the unit-entity table $\mathbf{E}$. This data organization sets subsequent analyses into a classical statistical framework, where the goal is to see how variables in $\mathbf{E}$ (the entities) affects the variables in $\mathbf{N}$ (the vocabulary) for each textual units. A choice was taken to use embeddings for analyzing these effects: units and words are embedded using Correspondence Analysis or pre-trained Word Embedding on $\mathbf{N}$, and entities are embedded in the same space as units using the Centroids or the Regressions methods on $\mathbf{E}$. These embeddings are then used in order to see affinities between entities and words, enabling the characterization of the formers by the latters. A case study on *Les Misérables* was performed to see if methods gave promising results.

The first important choice in the analysis is how to define the size of units. Other corpora were also tested (e.g. Shakespeare plays) and it seems important to define units with at least a paragraph size (after preprocessing) in order to represent them accurately. Choosing small units might be able to successfully capture word specificities related to a small subset of entities, but their direction becomes almost orthogonal one to another if the size of units is too small. This situation results in taking into account the "noise" in subsequent analysis, typically denoting an overfitting regime with an high variance and low bias. By contrast, large units will result in an underfitting regime, with a low variance and high bias, failing to capture entities specificities, but more robust to particularities in word usages. Having enough units is also important in order to properly locate entities in the embedding space. In order to analyze characters and relationships, we advice the practitioner to use his prior knowledge of the work in order to split the studied narrative as close as possible to "scenes" (as found in theater), which describes a particular event between an almost constant set of characters.

The second choice is to select which entities to study. This choice is of course driven by the studied problematic, but is also limited by the automatic extraction tools at disposal. These entities can be quite various, but must appear frequently in the work in order to be placed correctly in the embedding space. However, it is unadvised to set an entity which is almost always there (e.g. a narrator), as it will already be represented by the origin in $\mathbf{CENT}$ or as the constant term in $\mathbf{REG}$. As a rules of thumbs, the number of entities should ideally be lower than the number of textual units, however, even with an exceeding number of entities (like in our case study, where we had 547 entities for 365 units), if some entities appear rarely, analyses are still possible. Note that version of the

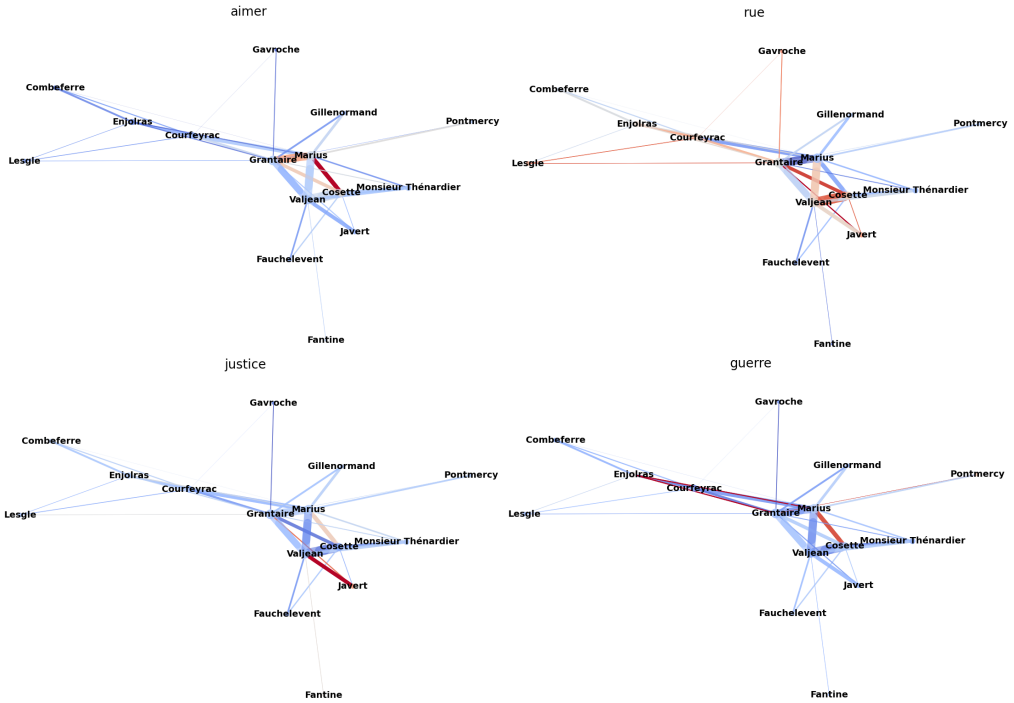

Figure 1: Resulting weighted and signed networks between main characters, with examples of word queries ("aimer", "rue", "justice", and "guerre"). These networks are computed with **CA-REG** method ($\lambda = 0.01$). Red indicate positive affinity, blue negative affinity, and edge width is proportional to the number of detected interactions between characters.

table $\mathbf{E}$ containing counts of entities rather than presence among each units was also tested in experiments, but gave similar result for the studied corpus.

The choice of using embeddings, also containing words, is motivated by the fact that the resulting space permit many types of exploration. As presented in this article, we can extract some of the most (or least) associated words with each entity or rank entities according to a word query. Nevertheless, other types of exploration could be made. Entities could be placed along a particular axis in the space, define with a set of positive and a set of negative words in order to highlight a particular contrast (good-bad, like in sentiment analysis, introvert-extrovert, friend-enemy, etc.). This approach could also be combined with a clustering of the words, or a Topic Modeling methods, thus permitting to define interpretable regions of the embedding space. Relative location of entities could also be used in order to cluster of classify them. All these leads could be explored by subsequent researches.

The difference in the choice between **CA** and **WV** embeddings appear quite clearly in the results. **CA** highlights particular words associated with entities, very specific to the studied work and the narrative events found in it, while **WV** give a feel of the general tone in the text found when these entities are used. This difference is explained by the fact that **CA** focuses on words appearing within the work, with possibly very different location to semantically similar words, while **WV** word-vectors are positioned regarding their semantic and syntactic similarities. A entity located in the **WV** space will then be in a semantic or syntactic region, and its characterizing words should all be related. Results show that **CA** methods generally perform better as a summary tools in order to understand the studied entities, but might bit limited in some kind of applications. As a matter of fact, the advantage of **WV** embedding is that it contains a larger vocabulary and that it is unchanging space regarding the different works. It should be prefered in order to compare different texts sequentially, and can be used when a list of word queries, which do not necessarily appear in every books, is used.

The choice between the **CENT** method and the **REG** method is relatively easy: thanks to its hyperparameter λ, the **REG** method can give similar results to the **CENT** method when λ is high (with only a contraction of entity vectors), but also give more "perpendicular" word affinities between entities when λ is low. Thus, it is clearly a superior choice in order to give a variety of results. The choice for this hyperparameter λ depends on what the practitioner want from his results. If her/his entities are

	V_1	V_2	V_3	V_4	V_5
CA-REG	huissier (1.4)	cuirassier (3.38)	gamin (1.92)	émeute (1.48)	sable (3.04)
	hôte (1.11)	infanterie (2.9)	mine (1.38)	révolte (0.86)	berge (2.32)
	arras (0.92)	sacrement (2.69)	farce (0.81)	bourgeoisie (0.84)	égout (2.16)
	lampe (0.91)	brigade (2.41)	ignorance (0.74)	populaire (0.82)	voûte (1.9)
	montreuil (0.81)	division (2.4)	jondrette (0.7)	insurrection (0.81)	vase (1.89)
	Valjean 1	Valjean 2	Valjean 3	Valjean 4	Valjean 5
	chandelier (1.03)	pelle (2.6)	ursule (1.49)	réverbère (0.98)	matelas (2.54)
	toulon (0.8)	fossoyeur (2.35)	luxembourg (1.06)	hausser (0.45)	ronde (1.96)
	gervai (0.73)	pioche (1.51)	tableau (0.93)	promenade (0.37)	galerie (1.06)
	bagne (0.65)	carte (1.39)	banc (0.81)	lanterne (0.36)	lanterne (0.99)
	maire (0.57)	mestienne (1.35)	mouchoir (0.77)	tuyau (0.35)	rive (0.99)
	Cosette 1	Cosette 2	Cosette 3	Cosette 4	Cosette 5
	gargote (0.59)	seau (1.48)	-	ravissant (1.11)	encre (0.76)
	balayer (0.57)	poupée (0.99)	-	céleste (0.76)	plume (0.59)
	alouette (0.56)	source (0.84)	-	volupté (0.67)	noce (0.49)
	servante (0.43)	gargote (0.71)	-	frémir (0.64)	chandelier (0.48)
	mois (0.34)	mestienne (0.64)	-	lancier (0.6)	antichambre (0.47)
	Cosette-Valjean 1	Cosette-Valjean 2	Cosette-Valjean 3	Cosette-Valjean 4	Cosette-Valjean 5
	maladie (0.49)	façade (0.68)	-	promenade (0.5)	noce (1.27)
	médecin (0.48)	corbillard (0.66)	-	chaîne (0.47)	marié (0.93)
	demain (0.33)	mestienne (0.6)	-	blessure (0.46)	mardi (0.89)
	surprise (0.28)	bâtiment (0.56)	-	tuyau (0.45)	mariage (0.86)
	auprès (0.28)	cul (0.55)	-	luxembourg (0.44)	file (0.65)

	V_1	V_2	V_3	V_4	V_5
WV-REG	demander (0.36)	saint (0.39)	gamin (0.45)	violence (0.44)	égout (0.52)
	décider (0.3)	mont (0.39)	garçon (0.42)	haine (0.42)	quai (0.45)
	aider (0.3)	régiment (0.38)	jeune (0.36)	révolte (0.42)	rue (0.44)
	expliquer (0.29)	chapelle (0.38)	enfant (0.35)	souffrance (0.4)	eau (0.42)
	plaindre (0.29)	infanterie (0.36)	père (0.34)	étincelle (0.39)	chaussée (0.42)
	Valjean 1	Valjean 2	Valjean 3	Valjean 4	Valjean 5
	essayer (0.29)	jean (0.54)	admirer (0.32)	jean (0.71)	jean (0.72)
	réfléchir (0.27)	jacques (0.33)	passer (0.31)	jacques (0.41)	pantalon (0.42)
	expliquer (0.24)	pantalon (0.29)	observer (0.3)	pantalon (0.36)	jacques (0.39)
	agir (0.24)	mr (0.28)	guetter (0.3)	louis (0.34)	philippe (0.34)
	questionner (0.24)	denis (0.28)	croiser (0.3)	philippe (0.33)	denis (0.33)
	Cosette 1	Cosette 2	Cosette 3	Cosette 4	Cosette 5
	an (0.41)	dormir (0.33)	-	rêver (0.32)	rêver (0.31)
	mois (0.39)	regarder (0.29)	-	regarder (0.31)	mentir (0.29)
	mère (0.36)	sentir (0.28)	-	contempler (0.28)	écrire (0.29)
	fille (0.35)	endormir (0.28)	-	pleurer (0.27)	demander (0.28)
	enfant (0.33)	respirer (0.28)	-	lire (0.27)	pleurer (0.28)
	Cosette-Valjean 1	Cosette-Valjean 2	Cosette-Valjean 3	Cosette-Valjean 4	Cosette-Valjean 5
	voir (0.32)	rue (0.55)	-	jean (0.52)	mariage (0.42)
	entendre (0.31)	ruelle (0.48)	-	pantalon (0.35)	marié (0.4)
	frissonner (0.3)	boulevard (0.45)	-	gilet (0.28)	noce (0.4)
	grommeler (0.3)	mur (0.4)	-	gris (0.27)	gai (0.33)
	essayer (0.3)	faubourg (0.4)	-	manteau (0.26)	amour (0.3)

Table 5
The 5 most associated words (similarity in parentheses) vs volumes constant, Valjen, Cosette, and Cosette-Valjean, as found in each tome regarding the **CA-REG** and **WV-REG** methods ($\lambda = 0.01$).

defined that some of them are completely included into other, such as character and character-pair, and she/he would like to have specificities about the finer grained entity, the λ must be set to a low value. By contrast, if he do not mind to have some of her/his entities defined as a mixture of others, he can set λ to a high value. However, very low values of λ should be avoided if the number of entities is high compared to the number of units, as this will lead to an overfitting of regression coefficients and will result in the association of very rare and specific

words with entities.

Finally, the biggest weakness of this framework yet is the difficulty to validate its pertinence. Several other case studies, with results carefully scrutinized regarding prior knowledge, should be undertaken in order to see if it result are solid, but this type of experiments are expansive both in time and explanation length. Another idea could be to use an annotated corpora such as in (REF), where human annotators already classified character relationships in various categories, and to see if the presented method can actually retrieve the different categories. Such an experiment can be promising, but it currently need highly performative automatic tools in order to detect and unify character entities occurrences, which is currently missing. Nevertheless, first case studies gave promising results for this framework, and its flexibility could lead to various applications.

A. Appendix

A.1. Correspondence Analysis

Starting from the $(n \times v)$ contingency table $\mathbf{N} = (n_{ij})$, we define the vector of unit weights as $\mathbf{f} = (f_i) := (n_{i\bullet}/n_{\bullet\bullet})$ and the vector of word weights as $\mathbf{g} = (g_j) := (n_{\bullet j}/n_{\bullet\bullet})$, where $\bullet$ denotes the summation on the replaced index. It is then possible to compute the *weighted scalar product matrix between units* $\mathbf{K} = (k_{ij})$ with

$$k_{ij} := \sqrt{f_i f_j} \sum_{k=1}^{v} g_k (q_{ik} - 1)(q_{jk} - 1), \tag{5}$$

where $q_{ik} = \frac{n_{ik} n_{\bullet\bullet}}{n_{j\bullet} n_{\bullet k}}$ is the *quotient of independence* of the cell i, k. The vector of textual unit i, $\mathbf{x}_i = (x_{i\alpha})$, is obtained by the eigendecomposition of the matrix $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ and with

$$x_{i\alpha} := \frac{\sqrt{\lambda_\alpha}}{\sqrt{f_i}} u_{i\alpha}, \tag{6}$$

where λ_α are the eigenvalues contained in the diagonal matrix $\mathbf{\Lambda}$ and $u_{i\alpha}$ the eigenvectors components found in $\mathbf{U}$. We find the vector of word j, $\mathbf{w}_j = (w_{j\alpha})$, with

$$w_{j\alpha} := \frac{1}{\sqrt{\lambda_\alpha}} \sum_{i=1}^{v} f_i q_{ij} x_{i\alpha}. \tag{7}$$

Note that various other quantities of interest can also be computed in CA, such as

$p_\alpha := \dfrac{\lambda_\alpha}{\lambda_\bullet}$: the proportion of inertia expressed in α,

$c^u_{i\alpha} := \dfrac{f_i x^2_{i\alpha}}{\lambda_\alpha}$: the contribution of unit i to axis α,

$c^w_{j\alpha} := \dfrac{g_j w^2_{j\alpha}}{\lambda_\alpha}$: the contribution of word j to axis α,

$h^u_{i\alpha} := \dfrac{x^2_{i\alpha}}{\sum_\alpha x^2_{i\alpha}}$: the contribution of axis α to unit i,

$h^w_{j\alpha} := \dfrac{w^2_{j\alpha}}{\sum_\alpha w^2_{j\alpha}}$: the contribution of axis α to word j,

For a detailed interpretation of these different quantities, see (REF).

A.2. Unit embedding based of pre-trained word vectors

This methods is justified and detailed in (REF). Let $\mathbf{w}_1, \ldots, \mathbf{w}_v$ be pre-trained word vectors which appear in the studied corpus, and the $(n \times v)$ table $\mathbf{N}$ counting the frequency of these words in the n textual units. We first construct the *uncentered vectors* $\widetilde{\mathbf{x}}_i$ of each unit i with

$$\widetilde{\mathbf{x}}_i = \sum_{j=1}^{v} \frac{n_{ij}}{n_{i\bullet}} \frac{a}{a + \frac{n_{\bullet j}}{n_{\bullet\bullet}}} \mathbf{w}_j, \tag{8}$$

where $a > 0$ is an hyperparameter which gives less importance to frequent words as $a \to 0$. In this article, we set a to the recommended value of 0.01. Let $\widetilde{\mathbf{X}}$ be the matrix whose columns are vectors $\widetilde{\mathbf{x}}_i$, and $\mathbf{u}$ be its first singular vector. We compute *vectors* $\mathbf{x}_i$ of each units i with

$$\mathbf{x}_i = \widetilde{\mathbf{x}}_i - \mathbf{u}\mathbf{u}^\top \widetilde{\mathbf{x}}_i. \tag{9}$$

This last equation act like a *centration* of unit vectors in the direction of the first singular vector $\mathbf{u}$.

References