Refining character relationships using embeddings of textual units

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

Distant reading tools allow researchers, from various fields, 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. Methods in this case rely heavily on Big Data and make an extensive use of Machine Learning algorithms. In the second case, researchers use these methods to underline hidden structures in a particular text, helping them to refine their understanding of it and reinforce stated hypotheses. Methods in this setting can also rely on Machine Learning, but must typically be build with more caution and attention to details: corpus are smaller, analyses are closer to the work, and methods must be more transparent in order to appropriately interpret results.

Automatic extraction and analysis of *character networks* from literacy works typically belong in the latter group. These methods aim at representing various interactions occurring between fictional characters found in a textual narrative with a graph, thus showing explicitly the hidden structure of character relationships constructed by the author. This structure might allow to find hidden patterns within a book, which can highlight a particular genre or style.

2. Methodology

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. Usually, the 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. In this paper, we propose a data organization leading to various methods which takes into consideration the text contained in the unit, helping to refine understanding of characters and their relationships.

2.1. Data organization

In this article, the data representation for a textual narrative, divided in n textual units $u_1, \dots, u_n$, is made through two tables. The first one is well known in the field of textual analysis, and consists in the $(n \times v)$ contingency table $\mathbf{N}$, as represented by Table 2.1, 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, denoted by $\mathbf{O}$, has a size of $(n \times p)$ where p is the number of *character-based objects* found in the narrative and cells o_{ij} counts occurrences of object j in unit i. A character-based object can be loosely defined in order to be flexible for various types of narrative or analyses, but can roughly be seen as a countable recurring narrative entity, containing one or more characters. For example, it can be a character occurrence, a character co-occurrence, an oriented character interaction (e.g.

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)

	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.

a dialog), or even a particular recurring event containing multiple characters (e.g. a meeting). In this article, we mostly consider character occurrences and pair of characters co-occurrences, as shown in Table 2.1.

	Cosette	Thénardier	Valjean	...
u_{101}	66	78	0	...
u_{102}	21	26	0	...
u_{103}	12	25	0	...
u_{104}	12	0	5	...

	...	Cosette-Thénardier	Cosette-Valjean	...
u_{101}	...	66	0	...
u_{102}	...	21	0	...
u_{103}	...	12	0	...
u_{104}	...	0	5	...

Table 2
A snippet of the character-based objects table $\mathbf{O}$ extracted from *les Misérables*. Rows are chapters, columns are character occurrences (top) and character co-occurrences (bottom), and cell o_{ij} counts the number of times object j appear in chapter i. Character co-occurrences are computed here as $o_{ij} = \min(o_{ik}, o_{il})$, where $\{k, l\}$ are characters constituting j.

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 object table $\mathbf{O}$ has an influence over the contingency table $\mathbf{N}$, i.e. searching which words are over-represented or under-represented knowing the character-based objects 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 in the textual unit act on her/his choice of words. If the extraction method permits it, a practitioner should include all character-based objects which she/he desire to study. Here, for example, the choice to include character co-occurrences along with occurrences is motivated by the fact that we are interested in studying character relationships. A character co-occurrence can roughly be seen as an interaction between two characters, and an interaction between two characters (or more, but higher order interactions are outside the scope of this article) should be considered as an object of its own: this interaction is not necessarily the sum of its parts and can give a particular flavor to the unit.

This data organization also highlight 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 character-based objects. 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 character-based objects, 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 two methods of analysis based on this table.

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 Sentiment Analysis (REF), Textometry (REF), or even Deep Learning methods (REF?). 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 (REF), these vectors of textual units are used as anchor points in order to also embed character-based objects 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 extract information about the localization of character-based objects vectors (the relative position of these vectors is insufficient). For that reason, we focus on embeddings of textual units which also contain *vectors of words*: by examining the positions of character-based objects relatively to word vectors, these objects can be characterized.

We propose here 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 in order to analyze textual resources has a long tradition in literature (REF). It has the advantage to naturally provide an embedding space, the factorial map, where units are placed alongside word vectors, allowing to analyze or represent the different units in terms of word frequency profiles. Moreover, units and words placement in the embedding space have a direct interpretation in terms of chi2 distance between profiles, which is desirable when interpreting results.

By performing a Correspondence Analysis on table $\mathbf{N}$, we directly get n vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ corresponding to units (rows) and v vectors $\mathbf{w}_1, \ldots, \mathbf{w}_v$ 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 position of a unit vector $\mathbf{x}_i$ can be interpreted by mutliple approaches, but we mainly use two here.

The first way to interpret it is to look at the unit-vector position on a particular axis α, i.e. looking at the component $x_{i\alpha}$. This component gives the position of the unit on a particular found "contrast" between units, depending on the chosen α, and this contrast will be strong if p_α, i.e. *the proportion of inertia expressed in* α, is high. It can further be interpreted by observing the *contribution of each word* j composing the axis α, i.e. $c^w_{j\alpha}$ and looking at the sign of $w_{j\alpha}$. Hopefully, by extracting and interpreting the most positive and negative contributing words, the chosen axis can express a particular duality which operates inside the studied text (e.g. love-hatred, action-description, lightness-darkness). The second method is to directly look at the *similarities* s_{ij} between an unit i and a word j, as defined by the scalar product between their vectors $s_{ij} := \mathbf{x}_i^\top \mathbf{w}_j$. 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, both methods are rather applied to character-based object vectors defined in section (REF), as they lie in the same space as unit vectors.

Using CA to embedded units however comes with some limitations. First, note that vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ obtained from CA reflect textual unit profile (in terms of words) regarding the mean profile. This analysis is therefore *contrastive*: it highlights unit variations among the studied text. This could becomes problematic in order to study a dataset containing multiple texts: units belonging to the same text will have a greater chance to be close from each other, and the displayed variance might be largely composed by differences of style between text. Another limitation with this approach is that the words helping the interpretation of units (and character-based objects) are contained in the studied text. Approaches requiring to study the position of units and objects relatively to a predefined list of words (e.g., friends, enemies, family) might therefore be impossible.

2.2.2. Pre-trained Word Vectors (WV)

Pre-trained word vectors, based on 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 quantity 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 mutliple applications (REF).

There exists various ways to use these pre-trained word vectors in order to find vectors for a *group of words*, such as sentences (REF), paragraphs (REF), or documents (REF). These vectors are often used to apply a classification or clustering algorithm on the newly embedded objects (REF), or to query information (REF). These methods often use the frequencies of words found 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 align vectors. In the present article, we use a recent methodology proposed in (REF) as it is compatible with multiple unit size and gives state-of-the-art results in multiple tasks.

More precisely, we use a pre-trained word vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ trained on Common Crawl and Wikipedia using fastText.[1] For French, the number of word-vectors m is 2 millions and the dimension of vectors are 300. Textual units vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ 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 $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (and character-based objects vectors in section REF) lie in an *absolute space* defined by the used 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 possible that all units from a given text, if the author style is particularly marked, are located in the same region of the space. In this case, the list of most associated word-vectors might be similar for every unit, and the analysis might not give usable results. This effect is fortunately limited by the centration of unit-vectors which occurs in the method describe in Appendix A.2.

[1] https://fasttext.cc/docs/en/crawl-vectors.html, accessed September 2022.

2.3. Character-based object embeddings

The main goal of this article is not to analyze units, but rather the character-based objects, i.e. the p columns of table $\mathbf{O}$. While we used the table $\mathbf{N}$ to build our embeddings of units, we now use the table $\mathbf{O}$ in order to build the character-based vectors $\mathbf{y}_1, \ldots, \mathbf{y}_p$ which lie in the same space as units.

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 proposed embeddings of units.

2.3.1. Centroids (CENT)

This method is the most trivial and is based on the following intuition: a character-based object 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

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

i.e. the center of mass, or *centroid*, of the units containing the character-based object, weighted by the relative weight of its occurrences $\frac{o_{ik}}{o_{\bullet k}}$. This way of building character-based vectors is closely related to the treatment of *supplementary variables* found in CA (REF): these variables do not act in the choice of factorial axis, but can still be represented afterward. However, by contrast, character-based 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 character-objects vectors positions are *additive*, which means that we have

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

where $\mathcal{G}$ is a subset of character-based objects. This property can be interpreted as followed: if the occurrences of 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 this implies that the specificities of the character might be hidden if he is often registered in an interaction. For example, if the character is frequently with a friend, the character vector will be close to the vector of the couple (if both the character and the couple are counted in the table $\mathbf{O}$). 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 to capture every specificities.

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 character-based object in a textual unit: the presence of a character a has a effect on the vocabulary of an unit, the presence of another character b has an other effect, and the presence of the pair $\{a, b\}$ yet a different effect. Now, the dependent variable still need to be defined. In fact, we are doing d regressions, with d the number of dimensions of the chosen unit embeddings, and each regressions is constructed to predict the α-th coordinate of each units by using variables in the $\mathbf{O}$. In a matrix notation, all models can be written as

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

where $\mathbf{X} = (x_{i\alpha})$ is the $(n \times d)$ matrix containing unit-vectors (on rows), $\widetilde{\mathbf{O}}$ the $(n \times (p+1))$ matrix representing $\mathbf{O}$ with an 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 $\mathbf{E}$ the $(n \times \alpha)$ error matrix.

Estimates $\widehat{\mathbf{B}} = (\widehat{\beta}_{k\alpha})$ for the intercept and coefficient regressions are in fact the researched embeddings for character-based objects 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 character-objects 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

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

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 $\mathbf{Y} \approx \frac{1}{\lambda} \widetilde{\mathbf{O}}^\top \mathbf{Diag}(\mathbf{f}) \mathbf{X}$, 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 character-object should be considered independently (with $\lambda \to 0$), to the hypothesis of additive mixture between character-objects (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 pertinent information about all character-objects.

3. Case studies

3.1. Datasets

4. Conclusion

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

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

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

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

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

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

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