JADT 2026, PROCEEDINGS, VOL II, P 655-664. EDITORS: ANTONELLA PLAIA, MARIANGELA SCIANDRA, ALESSANDRO ALBANO, UNIVERSITY OF PALERMO. HTTPS://JADT2026.UNIPA.IT

Validation of Simultaneous Additive Trees in Textual Data Visualization

Ludovic Lebart

CNRS (*R*) – ludovic@lebart.org

Abstract

Simultaneous (or augmented) additive trees are used to visualize lexical similarities and thematic relationships within large corpora. They allow words and texts to be represented on the same graphical representation, along with external (or supplementary) categorical variables describing these texts. In poetry, for example, the most frequent categories are collections, periods of creation, or major themes recognized by experts. However, the validation of observed patterns remains an insufficiently addressed problem. Interpretations often rely on visual inspection and heuristic criteria, which can limit their analytical reliability. The categorical variables mentioned above can be replicated by bootstrapping (a bootstrap replication being a random sampling with replacement of their texts). Replications of size n (e.g., n = 50) allow the calculation of confidence ellipses (or convex confidence hulls) for the position of the categories. The methodology is illustrated by an application to a classic literary corpus: the 154 sonnets of William Shakespeare (with their classification into eight themes, recognized by many experts). This corpus will serve as a reference for defining and comparing certain characteristics of the proposed validation procedure. A complete, transparent set of Python programs (with a user-friendly Tkinter interface) will be freely available from the websites: www.dtmvic.com and https://ludoriv.github.io/DTM-VIC.

Keywords: Simultaneous Additive Trees, Bootstrap validation, Visualization. Poetry, Shakespeare Sonnets.

1. Introduction

Visualizing textual data here involves producing graphical representations that highlight proximities between textual units (words, graphic forms, lemmas) and proximities between texts within the same corpus (speeches, novels, poems, authors, etc.). Correspondence analysis (CA) is a widely used tool, particularly because of its ability to simultaneously represent textual units and texts. Another descriptive tool is additive trees (AT) (or phylogenetic trees), which attempts to represent calculated distances in spaces of more than two dimensions. The simultaneous representation of texts and units, and the positioning of additional variables, are also possible with AT, although they do not offer the optimal performance of CA. In this work, we show that in both AT and CA, non-parametric confidence zones for point positions can be plotted. This technique will complement CA, not replace it.

Correspondence analysis (CA) remains an essential step for initial work with lexical tables, but it does not guarantee a comprehensive overview. In contrast, additive trees, with their interpretation rules, allow for the reconstruction of distances in a plane (a tree is a planar graph) within spaces of dimensions far exceeding two. However, they lacked the simultaneous representation of rows and columns (i.e., words and texts) of the basic lexical table. A pragmatic

method for the simultaneous representation and positioning of additional variables was proposed at JADT2024 in Brussels (Lebart, 2024). It is summarized in Section 2. A non-parametric statistical validation procedure based on the bootstrap technique is presented in Section 3. Section 4 then describes the text corpus (Shakespeare Sonnets) used as an illustration, while Section 5 presents the application itself. Finally, Section 6 provides some pieces of information about the self-contained Python program chain.

2. Correspondence Analysis (CA) and Additive Trees (AT)

The calculation of additive trees, originally proposed by Buneman (1971), was improved by Sattath and Tversky (1977). The work of Barthélémy and Guénoche (1988) and Luong (1988) promoted the use of these methods (under the name of "*Analyse Arborée*") in the field of text analysis. Additive trees were made easily computable by Saitou and Nei (1987) and subsequently became established as a synthesis of principal component analysis and clustering techniques (hierarchical classifications, k-means partitions, etc.). Bryant (2005), and then Huson and Bryant (2006), later justified the method and made corresponding software available. Theoretical justifications for the algorithm's effectiveness were presented by Mihaescu et al. (2009).

Let us briefly recall that the concept of hierarchy at the heart of hierarchical clustering amounts to approximating the initial distances by an ultrametric distance, which, in addition to the classical axioms of distances, satisfies the following inequality for any triplet (*x, y, z*):

$$d(x, y) <= Max\ (d(x, z), d(y, z)\ )$$

Additive trees require, for any quadruplet *(x, y, z, t)*, that the following more complex and less intuitive, but less demanding, inequality be satisfied:

$$d(x, y) + d(z, t) <= Max\ (\ \{d(x, z) + d\ (y, t)\}, \{d(x, t) + d(y, z)\}\ )$$

A tree can then be drawn with the objects to be classified as terminal elements (or "leaves"). More flexible than the minimum spanning tree, which depends on *n-1* parameters, the additive tree involves *2n-3* parameters. It is therefore necessary to find an approximation of the initial distances that satisfies these conditions, which is made possible by the mentioned work of Saitou and Nei. The important result is that the distances between nodes (texts) of the graph (length of the shortest path on the graph) are approximations of the actual distances throughout the whole space, approximations often better than those provided by a first AC plane.

2.1 On the options for drawing additive trees

It remains to draw these trees. For a comprehensive review of general graph visualizations, see Di Battista *et al.* (1999), and more specifically, regarding methods using force-directed drawing algorithms, the article by Kobourov (2013), which analyzes more than 60 publications corresponding to several dozen algorithms. Tutte's graph drawing algorithm (1963) was one of the first drawing methods based on algorithms of this type. Then, the methods proposed by Eades (1984) and the Fruchterman and Reingold algorithm (1991) both rely on repulsive forces between all the nodes of the graph, but also on attractive forces between adjacent nodes (the edges are considered springs, and the goal is to find a balance between all the tensions, hence the name force-directed drawings).

Alternatively, the forces between nodes can be calculated based on concepts from graph theory. The distances between nodes are then the lengths of the shortest paths connecting them. For additive trees, these distances are precisely an approximation of the original distances (chi-square distances calculated from the original lexical table). The algorithm of Kamada and Kawai (1989) uses these "spring forces" proportional to these distances calculated from the graph. This plotting algorithm is therefore the most compatible with the properties of additive trees, and thus with the basic lexical distances. Experimentally, the good compatibility of these representations with the principal planes derived from the CA of the original lexical table is also observed. However, a validation technique remained necessary to confirm that observation.

The ability of correspondence analysis to describe undirected graphs of the planar grid or geographical map type from their associated matrices has been highlighted by Lebart *et al.* (1998). One might have thought of using correspondence analysis again to obtain a plot of the additive tree. However, trees (connected graphs without cycles) are poorly visualized by this method, and vice versa, as noted by Sattath and Tversky (1977): *"It is interesting to note that tree and spatial models are opposing in the sense that very simple configurations of one model are incompatible with the other model. For example, a square grid in the plane cannot be adequately described by an additive tree."*

Finally, we proceed with the representations provided by the Kamada-Kawai method, the most suitable for additive trees, enrich them with a simultaneous representation of the rows and columns of the lexical table, and complement them with a bootstrap validation procedure.

2.2 Simultaneous Representation and Characteristic Words

CA can easily be presented directly as the search for the best possible simultaneous representation of the proximities between rows and columns of a contingency table. One could indeed look for a simultaneous positioning of texts and words on an axis (to begin with) in order to obtain a doubly centroidal relationship: words at the centroid of the texts, and texts at the centroid of the words (the weights being respectively the lexical profiles in rows and columns calculated on the basic lexical table). This double relationship is impossible, because centroid computation is contractive: words must be within the interval covered by the texts and, simultaneously, the texts must be within the interval covered by the words. For the relationship to be possible, these centroids must be expanded (coefficient $\beta_\alpha > 1$, for axis α).

The optimal solution corresponds to values of β_α closest to *1*, which gives us the positions of the words and texts on the first axes of the correspondence analysis. These doubly barycentric relations are known as "transition relationships".

Therefore, we cannot hope to find an optimal dual simultaneous representation, but a single simultaneous representation is sufficient (words as centroids of the texts, which are the terminal elements of the tree). We will enrich this simultaneous representation by also incorporating the notion of characteristic words (or specificities).

The simultaneous representation procedure includes the following steps: (columns and rows play similar roles and can be interchanged).

- a) Preliminary CA of the lexical table.
- b) Choice of the dimension *nx* of the space deemed significant (generally by bootstrapping) (*12* axes, for example). Distances will be calculated from the first *nx*

principal axes of the CA (in our illustrative examples, *nx* = 12). This allows for the regularization of the initial distances, a well-known procedure in discriminant analysis and in certain Deep Learning procedures.

- c) Calculation of the additive tree (Neighbors-Joining method) on the distance matrix between the coordinates of the columns (texts) on the first *nx* axes.
- d) Plotting the tree (Kamada-Kawai procedure)
- e) The rows (words - forms, lemmas) are positioned barycentrically based on the coordinates of the column points (texts) (terminal elements of the tree) issued from Kamada-Kawai procedure and the textual profile of the rows (words, tokens, lemmas).

To limit the number of words on the graphs, the characteristic rows/words (with a fixed probabilistic threshold) are calculated directly from the lexical table for each column-text using test values (for test values, see, for example, Lebart *et al.*; 1998, 2019).

In practice, new edges (different in color and thickness from those of the edges of the additive tree) are drawn, joining each column point (text) on the graph to its characteristic rows (words).

The proposed methodology is illustrated by its application to a classic literary corpus: the 154 sonnets of William Shakespeare (with their classification into eight themes, recognized by experts). This corpus will serve as a reference for defining and comparing certain characteristics of the proposed validation procedure.

3. Validation and bootstrap

Computer intensive techniques allow us to go far beyond the criterion of interpretability of the results that was frequently used during the first phases of the upsurge of data analytic methods. To compute the precision of estimates, the classical analytical approach is both unrealistic and analytically complex. The bootstrap (see: Efron and Tibshirani, 1993), on the contrary, makes almost no assumption about the underlying distributions, and gives the possibility to master every statistical computation for each sample replication and therefore to deal with parameters computed through the most complex algorithms.

The nonparametric bootstrap consists in drawing with replacement K samples of size n out of n statistical units. Then, parameter estimates such as means, variances, coordinates on a specific plane (in our case) are computed on the K new obtained samples. Empirical evidence suggests that 30 is an acceptable value for K in the context of Principal Axes Methods. We have at this stage K samples (the replicates) drawn from a new theoretical population defined by the empirical distribution of the original data set, leading to K estimates of the parameters of interest.

In our case, we will use the Partial Bootstrap (see Greenacre, 1984), Chateau and Lebart, 1996): We do not question the validity of the coordinates of the active statistical units (words and texts) but we wish to appreciate the stability of the supplementary categories (topics in our example). One replication then consists of a drawing with replacement of the text units (the 154 sonnets in the case of the example of section 4 below). This leads to a vector of bootstrap weights, such as 0, 1, 2, 3… the sum of the weight being 154. A replicated category is then a weighted barycenter of the text units. The process is the same for CA and AT. The cloud of replicates for one category allows for the drawing of a confidence ellipse containing 95 % of the replicates. Examples of such confidence areas are given below for both CA and AT (Fig. 3 and 4).

4. Application: Corpus and context

4.1 An Overview of the Content of Shakespeare's Sonnets

William Shakespeare's 154 sonnets deal with themes or topics such as love, friendship, the effects of time, beauty, betrayal, lust, and death. Three contiguous sets of sonnets are generally recognized as corresponding to three dominant themes:

Sonnets 1-17: (Procreation). These sonnets celebrate the beauty of a young man whom the poet urges to marry in order to perpetuate that beauty.

Sonnets 18-126: (Young Man). This sequence, the longest, concerns the same young man (not identified), the destructive effect of time, and the power of love, friendship, and poetry.

Sonnets 127-154: (Dark Lady). These sonnets are primarily addressed to a dark-haired woman.

The themes Young Man and Dark Lady could themselves contain five sub-themest assigned to the five new categories below (Absence, Storm, Rivalry, Death, Eternal Poetry).

Table 1. List of eight a priori categories (topics) with corresponding sonnet numbers

Category 1: Procreation 1 - 17
Category 2: Young Man 20-25, 33-38, 40-42, 46, 47, 49, 53-55, 59-60,62-70, 75-77, 88-106, 108-112, 115-125,
Category 3: Dark Lady 127-136, 139, 140, 143-146, 153,154
Category 4: Absence 26-32, 39, 43-45, 48, 50-52, 56-58, 61, 113-114
Category 5: Storm 141,142,147-152
Category 6: Rivalry 78-87
Category 7: Death 71-74
Category 8: Eternal poetry 18, 19, 81,107

The sonnet divisions in Table 1 are inspired by the work of Alden (1913) and Paterson (2010).

4.2 Specificity of poetic texts

The statistical analysis of poetry and song collections is hampered by the specific nature of these texts. They may contain choruses, deliberate repetitions, stylistic devices, or versification constraints that challenge the statistical significance of lexical frequencies. A pragmatic solution is to work with "bags of words" (the presence or absence of words, tokens, lemmas) which constitute only a tiny fraction of the texts' artistic aspect, but nevertheless prove useful for studying, for example, a work within its temporal context. This is not an in-depth study or an elucidation of poetic texts, but simply an examination of lexical traces, which will ultimately prove to be of interest.

4.3 Inadequacy of Principal Axes

With the presence-absence coding of words in text units, frequencies are somewhat constrained. Word bags are created from a certain size of text units. These units will be the poems themselves in the case of poetic works. In our case, for Shakespeare, the text units will be the 154 Sonnets. The maximum frequency of a word cannot exceed the number of text groups, since it is simply a matter of "presence-absence" coding within each text unit. Under these conditions, the dimensionality and sphericity of the word cloud rarely allow for good visualizations in a low-dimensional space. In the case of CA, the eigenvalues will be poorly separated, a source of

instability for the principal axes. The two-dimensional representation of words and texts, an undeniable advantage of CA, becomes insufficient, and additive trees become indispensable.

5. Implementation and examples of visualization

We will characterize and summarize our approach with the help of four graphical displays.

5.1 Figure 1, Aggregated lexical table

For Figure 1, the eight themes are considered known and indisputable. They were used to construct the lexical table (Themes x words). Figure 1 then shows the simultaneous additive tree (SAT) linking the eight themes, which are terminal elements of the tree (its "leaves"), with the positioning of the words (which are centroids of the themes). The words were linked to the themes they most characterize (criterion: test values). As with the following figures, we are dealing with vocabulary and not lexical frequency: each sonnet is a bag of words. The themes play an active role in the construction of figure 1. This figure reminds us of the principle of the SAT.

5.2 Figure 2, Direct Additive Tree

The additive tree in Figure 2, on the other hand, is derived from the AT of the lexical table (Sonnets x words) (the 154 sonnets being identified from "S01" to "S154"), It completely ignores the themes. These themes are then located *a posteriori*. As in CA, they are centroids of the implied sonnets, these centroids being expanded to equalize the dispersions of the sonnets and of the themes (the famous β coefficients mentioned earlier in section 2.2).
For example, we can see that the theme "Death" (at the top) is particularly far removed from the theme "Rivalry" (at the bottom right). Words could also be used to characterize the sonnets (with colors and a much larger figure format...).

5.3 Figure 3, Direct CA with confidence areas

Figure 3 shows the first principal plane of a CA of the same lexical table as Figure 2. The words are still omitted for readability, but the themes are shown as additional elements. Only two bootstrap confidence ellipses for the position of the themes are drawn (again for readability) for the two preceding themes (boxed), Death and Rivalry. We can see that the themes are not as far apart as in Figure 2, but also that their confidence ellipses have a fairly clear intersection, which casts doubt on the significance of their distinction.

5.4 Figure 4, Direct AT with confidence areas

Finally, Figure 4 shows us the same additive tree as in Figure 2, but now with bootstrap confidence ellipses for the two highlighted (and boxed) themes. It is clear that these themes are not only distinct but almost opposite, something the principal plane of the CA, limited to two dimensions, could not detect. One could argue that it is possible to observe such divergences by analyzing the sequence of principal axes beyond axis 2. However, practitioners know that this is difficult and tedious, as proximities in the (3,4) plane, for example, must be interpreted "all other things being equal" ("ceteris paribus") in the (1,2) plane...

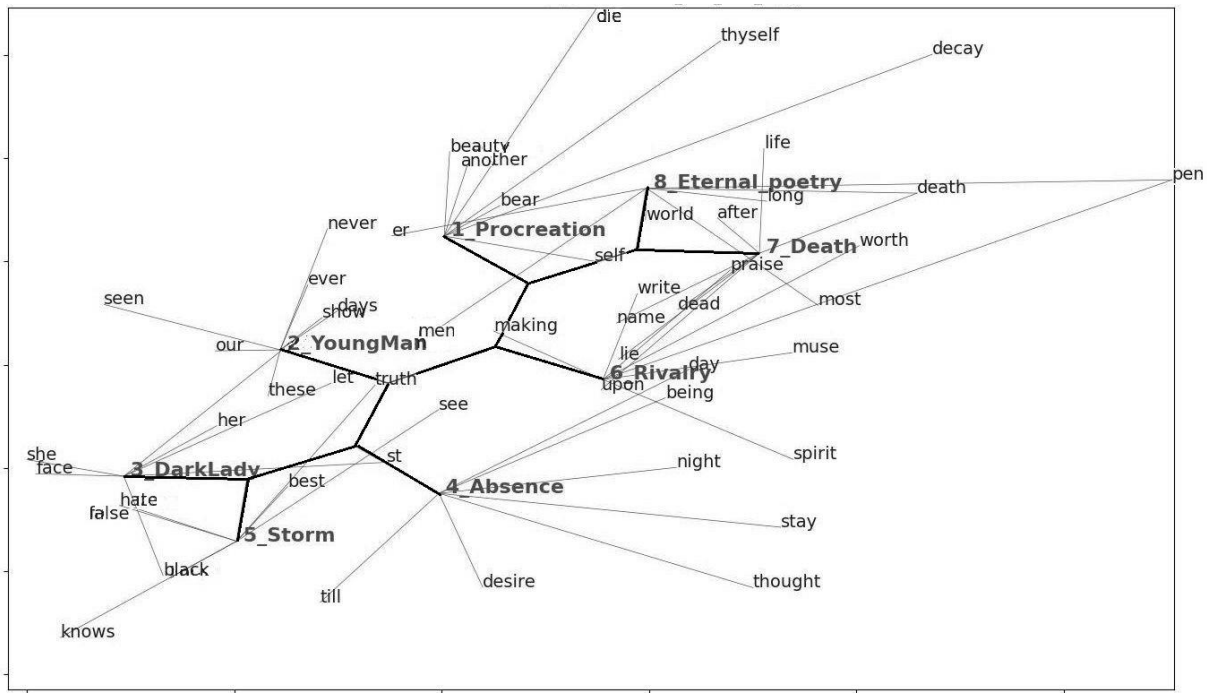

Figure 1. Simultaneous Additive Tree on the aggregated lexical table cross tabulating 8 categories (terminal elements of the tree) and 259 words (word frequencies > 9).

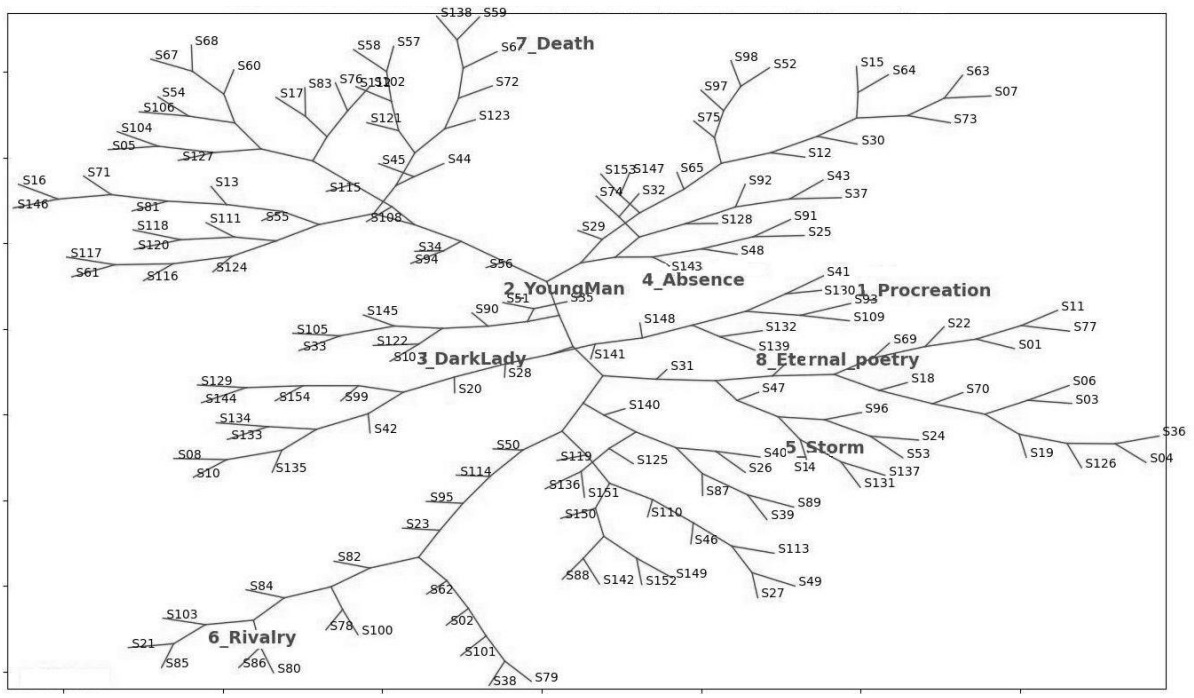

Figure 2. Simultaneous Additive Tree whose terminal elements are the 154 sonnets, with the 8 categories as supplementary elements. The data input is the lexical table cross tabulating 154 sonnets and the same 259 words. For better readability, the words are not represented here.

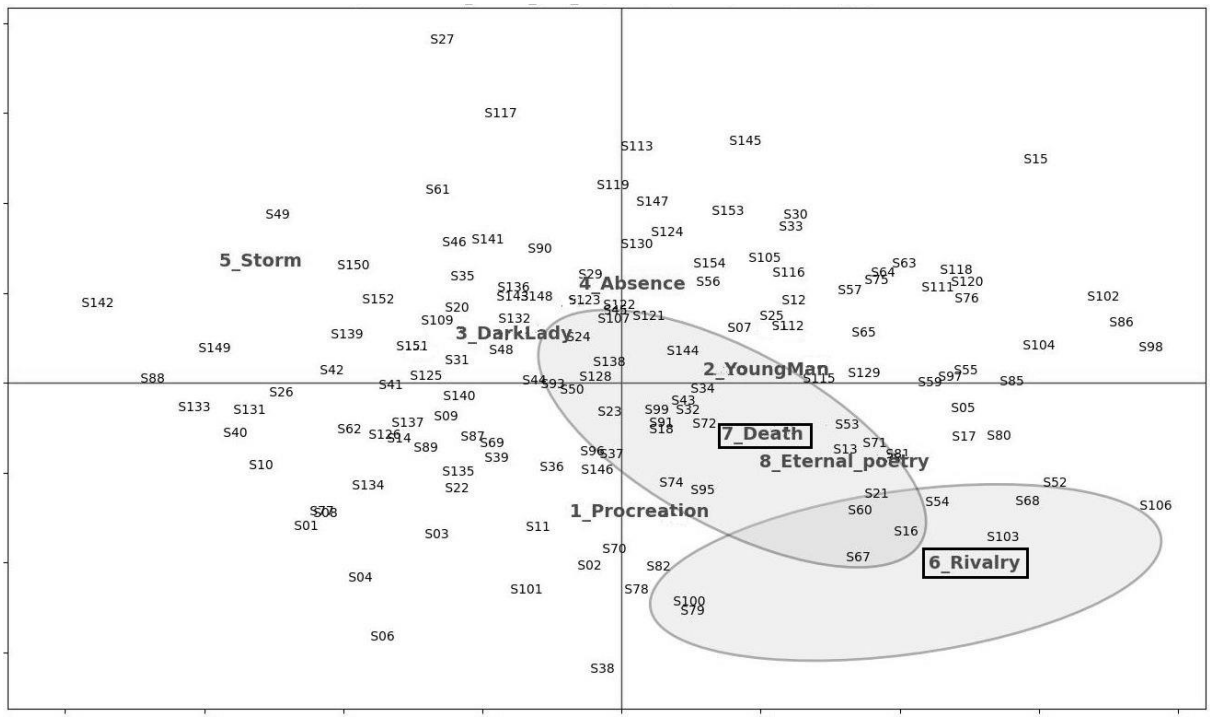

Figure 3. Plane (1, 2) from the CA of the same lexical table as in Figure 2. In this classical simultaneous representation, the 154 sonnets are identified by their traditional numbers in the sequence of sonnets. Two bootstrap confidence ellipses are drawn, for the two topics 7_Death and 6_Rivalry (squared identifiers). These ellipses are overlapping, inducing a disputable distinction between these two topics.

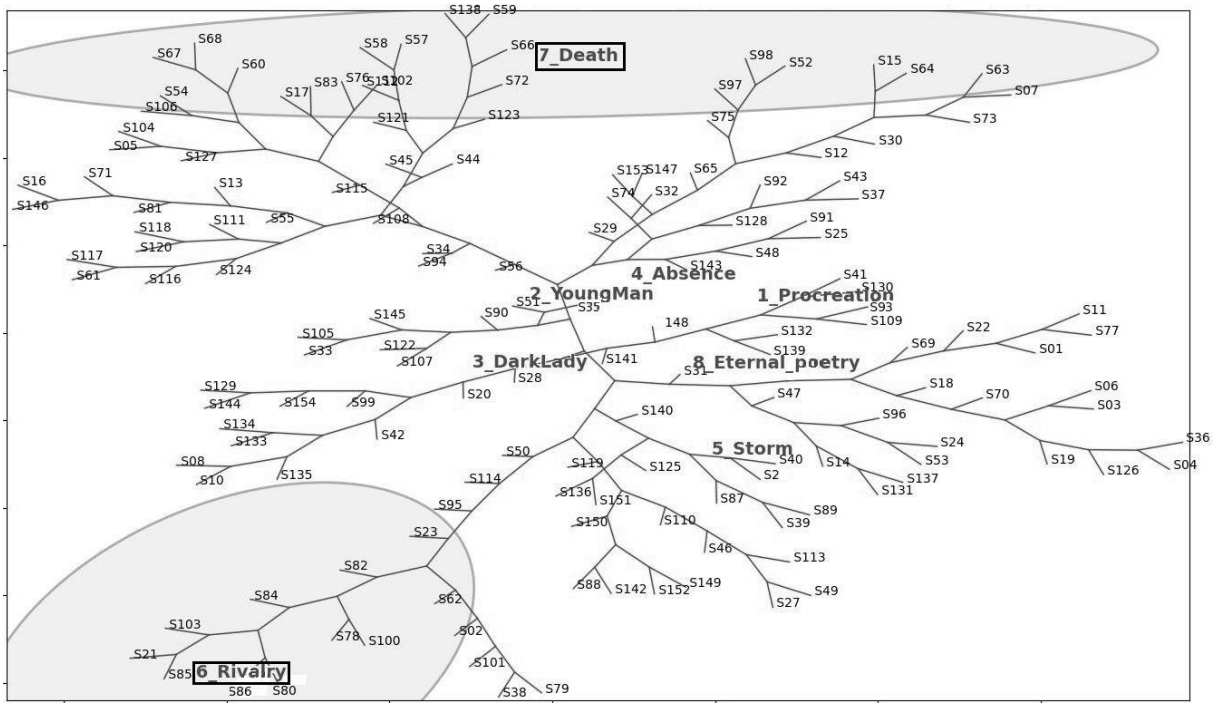

Figure 4. On the same tree as in figure 2, the confidence ellipses of the two topics Death and Rivalry are drawn, exhibiting now a marked difference between these topics, probably imputable to the axes of order >2.

6. Python Software

The Python code for the complete seven- or nine-step processing chain (from raw text to bootstrap confidence ellipses plotted on simultaneous visualizations of additive trees), will be freely available at both URL: www.dtmvic.com and https://ludoriv.github.io/DTM-VIC.

This chain uses the following standard Python modules: *Tkinter* (interface, graph framing), *os*, *Lib, numpy, pandas, matplotlib, igraph, importlib, textwrap, re* (regular expression), *sklearn*, and *scipy*.

The interface allows users to select folders, files, and parameters:

A first section: *Select folder* looks for the basic folder containing data files and modules.

In the section entitled: *Select files*:

"Select basic_text": (raw input text : tokens, lemmas, etc.)

"Select categories": (XL file (.csv) file describing categories of texts)

"Select identifiers": (XL file (.csv) file describing the identifiers of categories)

In the section: *Parameters*

"wordbag": If wordbag = 1: Each raw text is replaced with a bag of words; if wordbag = 0: the analyses are performed on raw texts,

"anagreg": If anagreg = 1: Poems/texts are aggregated according to the categories described in category file; if anagreg = 0: Direct analysis of all poems/texts, leading to validation procedures.

"freqmin": freqmin: Minimum frequency for kept words.,

"nax": nax: Maximum number of axes to keep in the initial correspondence analysis.,

"nvt": nvt: Number of kept characteristic words (criterion: test values)

"Detail": Detail = 1: For anagreg = 0, detailed outputs (bootstrap replications on graphical displays as a check, various visualizations); Detail = 0: basic outputs.,

"seuil_vt": seuil_vt: Significance threshold of the test-values (for selection of characteristic words).

7. Conclusion

Despite several similarities with artificial intelligence as regards some algebraic or algorithmic tools, data visualization methods occupy a particular place in a knowledge process. A visualization is neither a simple task to perform nor a decision to make. We submit our texts to understand and reflect. The approach is mainly unsupervised, a phase of work that Deep Learning will increasingly need, according to the predictions of the seminal text by LeCun et al. (2015). As in correspondence analysis with its simultaneous representations, the observable pattern of column points (texts, collections) tells us how these points are organized, and the presence of row points (words) and additional variables (themes) tells us why they are organized in this way: texts are similar because they often use the same words or deal with the same themes. It has been said that words and categories dress the skeleton of the additive tree. But this dressing would be merely a disguise in the absence of validation procedures. The non-parametric validation proposed here hopes to give these visualizations a scientific status.

References

Alden, R. M. (1913). Sonnets and a Lover's Complaint. New York: Macmillan.

Barthélémy J.-P. and Guénoche A. (1988). *Les arbres et les représentations de proximité*. Paris: Masson.

Bryant D. (2005). On the uniqueness of the selection criterion in Neighbor-Joining. Journal of Classification, vol. (22), 1: 3-16.

Buneman P. (1971). The recovery of trees from measurements of dissimilarity. In: Hodson F. R. D. Kendall G., and Tautu P., (Editors). Mathematics in the archeological and historical sciences. Edinburgh University Press, Edinburgh: 387-395.

Chateau, F. and Lebart, L. (1996). Assessing sample variability in visualization techniques related to principal component analysis: bootstrap and alternative simulation methods. In : A. Prats (Ed): COMPSTAT96, Physica Verlag, Heidelberg, 205-210.

Di Battista, G. Eades, P., Tamassia R., and Tollis, I.G. (1999). Graph Drawing: Algorithms for the Visualization of Graphs, Englewood Cliffs: Prentice. Hall.

Efron, B. and Tibshirani, R. J. (1993). An Introduction to the Bootstrap, Chapman and Hall: New York.

Eades P. (1984). A heuristic for graph drawing. Congressus Numerantium, 42:149–160.

Fruchterman T. and Reingold E. (1991). Graph drawing by force-directed placement. Softw. – Pract. Exp., 21(11):1129–1164.

Greenacre, M. (1984): *Theory and Applications of Correspondence Analysis*. Academic Press, London.

Huson D. H. and Bryant D. (2006) Application of Phylogenetic Networks in Evolutionary Studies. Molecular Biology and Evolution, 23 (2): 254 - 267. Software available from www.splitstree.org.

Kamada T. and Kawai S. (1989). An algorithm for drawing general undirected graphs. Inform. Process. Lett., 31:7–15.

Kobourov, S. G. (2013). Force-Directed Drawing Algorithms. in: Handbook on Graph Drawing and Visualization. Chapman and Hall/CRC.

Lebart L. (2007). Which bootstrap for principal axes methods? In: Selected Contributions in Data Analysis and Classification. P. Brito *et al.* (eds), Springer: 581-588.

Lebart L. (2024). Des outils pour décrire certains corpus de poèmes et de chansons : Les arbres additifs simultanés. (2024). In: JADT 2024, "Mots comptés, Textes déchiffrés" Anne DISTER et Dominique LONGREE (eds) Vol. 2, PUL, (Presses Universitaires de Louvain) : 553- 562.

Lebart L., Pincemin B. and Poudat C. (2019). *Analyse des Données Textuelles.* Presses de l'Université du Québec, Québec.

Lebart L., Salem A. and Berry L. (1998). *Exploring Textual Data,* Kluwer, Dordrecht.

LeCun Y, Bengio Y. and Hinton G. (2015). Deep learning. *Nature*. May 28;521(7553):436-44.

Luong X. (1988). *Méthodes d'analyse arborée. Algorithmes, applications.* Thèse pour le doctorat ès sciences. Université Paris V.

Mihaescu R., Levy D. and Pachter L. (2009). Why Neighbor-Joining works? Algorithmica, vol. (54): 1-24.

Paterson D. (2010). Reading Shakespeare Sonnets. Faber & Faber Ltd. London.

Saitou N. and Nei M. (1987). The neighbor joining method: a new method for reconstructing phylogenetic trees, Molecular Biology and Evolution, vol. (4), 4: 406-425.

Sattath S. and Tversky A. (1977). Additive similarity trees. Psychometrika, vol. (42), 3: 319-345.

Shakespeare, W. (1901). Poems and sonnets: Booklover's Edition. Ed. The University Society and Israel Gollancz. New York: University Society Press. Shakespeare Online. Dec. 2017.

Tutte, W. T. (1963). How to draw a graph. Proc. London Math. Society, 13(52):743–768.