Search for Meaning Through the Study of Co-occurrences in Texts

Nicolas Bourgeois, Marie Cottrell, Stéphane Lamasse, Madalina Olteanu

Search for meaning through the study of co-occurrences in texts

Nicolas Bourgeois[1], Marie Cottrell[1], Stéphane Lamassé[2], and Madalina Olteanu[1]

[1] SAMM - Université Paris 1 Panthéon-Sorbonne
90, rue de Tolbiac, 75013 Paris, France
nicolas.bourgeois,marie.cottrell,madalina.olteanu@univ-paris1.fr
[2] PIREH-LAMOP - Université Paris 1 Panthéon-Sorbonne
1, rue Victor Cousin, Paris, France
stephane.lamasse@univ-paris1.fr

Abstract. In this paper, we combine several tools used in text-mining in order to study both the lexicon and the semantic structure of a set of medieval texts. On the one hand, the study of occurrences (Principal Component Analysis, Topic Models, Self-Organizing Maps, Hierarchical Cluster Analysis) allows a wide scope of tools to extract and display information from big data. On the other hand, the study of co-occurrences (words belonging to a sentence, a paragraph) allows to keep track of the structure of each text, but is more tedious to handle and often leads to messy visualizations. Here we use the SOM algorithm to reduce the size of the data (clustering, removal of fickle information) while preserving the semantic structure ; then we can rely on classical but slower algorithms (HCA, graph representation) to purpose data visualization.

Keywords: Text Mining, SOM, Graphs

1 Introduction, state of the art

With the development of the WEB sites, social networks, big data depositories and real time information flows, text mining has become an important focus in several fields such as statistics and computer science. Many different techniques can be used to extract lexical and semantic information from texts of various sources (literary, technical, political, scientific, and so on). We will briefly recall two of them in this introduction : counting frequencies and topic modeling. Those two, actually most of the more popular ones also, consider each text as a simple bag of words and pay little attention to its structure.

On the contrary, the study of co-occurrences focuses on the articulation between words inside of a text : sentences, paragraphs, distances. We will recall the general principle of that technique, and its main drawback also which is the difficulty to display and analyse results.

The main goal of this paper is to adapt some tools that had been developed to improve display, robustness or efficiency of frequency analysis, in order to

cope with this drawback. In particular we hope to produce sound results on the analysis of co-occurrences for a corpus of scientific medieval texts.

1.1 Words frequencies

The most used and the most popular method consists in the computation of the absolute frequencies (or *number of occurrences*) for each word, eventually lemmatized [1]. Then the words can be sorted by decreasing value of their frequencies, so that each text is linked to the set of words which have the higher importance for it. In this case, each text is viewed as a bag of words, and the structuring into sentences or paragraphs is neglected. The data consist of a contingency table where entry $a_{i,j}$ is the number of occurrences of word i in text j. It is in this frame that The Factorial Component Analysis (see Benzecri [1992] or Lebart et al. [1984]) is used in order to provide simultaneous representations of both the words and the texts, allowing the study of the proximity between words, between texts, between texts and words.

This method is useful to put in evidence some clusters of words, some associations between texts and clusters of words, but it is impossible to get a global representation of these associations. That is due to the limitations of the Factorial Component Analysis projections, (see for example Bourgeois et al. [2015]) where we propose an improvement to overcome this drawback. But in any case, the semantic aspect is definitely lost.

1.2 Topic Models

Topic Model is a statistical model which assumes that there exists hidden groups of words, called *topics*, which are meaningful but that the observer cannot address directly. The actual texts we can observe are built through a process of picking words from one or several of these topics. This approach recalls somehow of the so-called Mixture Models (Titterington et al. [1985]).

The aim of the observer is to do some reverse engineering and to deduce the topics from the texts, which can be very different depending on the exact statistical model chosen. Popular methods in that field include PLSI or LDA (see Blei [2012]). In this frame also, the semantic disappears as each text is considered as a simple bag of words. Actually, since the reversion operation is usually tough compared to the generation process, a topic model is probably not the easiest way to study language structures.

1.3 Co-occurrences Analysis

The analysis of co-occurrences (Martinez and Salem [2003], Martinez [2012]), helps to cluster the words without breaking the links with semantic analysis. It

[1] Lemmatization consists in gathering the different inflected forms of a given word as a single item: for example *choose*, *chose*, *chosen* will be associated to the item *choose*. Note that the lemmatization is a much easier task for English language than for other languages which use the declension and the conjugation in their grammar.

deals with the concordances between words and not only with their number of occurrences.

The texts are split according to a given level of segmentation : sentences, paragraphs, sections, for example. In the following, we decide to use the *paragraphs*. All the words are considered, with or without lemmatization, including numbers, whatever the texts they belong to. The only excluded forms are the punctuation signs. All the texts are gathered into a unique text.

For each given word, called *pole* p, and for each other word w, we count how many times w is present in a paragraph which contains p. This number is the number of co-occurrences of w with the pole p.

For example, let us take the sentence :

> One *approach* to understand the *evolution* of science is the study of the *evolution* of the language used in a given field.

In this sentence, if *approach* is chosen as *pole*, the word *evolution* is its co-occurrent twice, while if we invert the role of *approach* and *evolution*, the word *approach* is co-occurrent of *evolution* once.

In the sequel, the main object of our study is the co-occurrence matrix C. It is a square matrix $T \times T$, where T is the total number of considered words (or lemmas if lemmatization is applied). Each word is seen as a pole as well as a co-occurrent. We denote by $n_{i,j}$ **the number of times where word j is inside a paragraph which contains word** i. In this notation, i is the pole, j is one of its co-occurrents. Note that this matrix is not symmetric, since according to the definition, $n_{i,j} \neq n_{j,i}$.

This co-occurrence matrix allows to define a weighted and directed graph where the words are the vertices and the values $n_{i,j}$ are the weights of the edges. For an elementary example, the reader may refer to Figure 1.3.

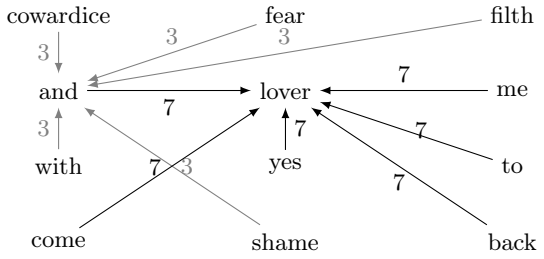

Fig. 1. Graph of co-occurrences from the 2-paragraphs quote "With fear and filth and cowardice and shame./ Yes and lover, lover, lover, lover, lover, lover, lover come back to me". For the sake of readability, edges of weight 1 have been omitted

This graph shows all the links between all the lexical forms (seen as pole or co-occurrent) and takes into account the context and the semantic structure of the

texts. A great difficulty appears immediately : this graph is usually unreadable and it is impossible to extract a meaningful information. Notice for example that every paragraph is a clique (a complete cluster) over all the words it contains. Hence the need to simplify the complete graph in order to decrease its complexity.

1.4 Towards a more compact and more readable graph

The aim of the manuscript is to look for a smaller graph, sparsely connected, where most of the useful information from the initial graph would still be preserved. This goal can be achieved by defining groups of strongly connected words that are subsequently taken as vertices and by studying the connections between these *new* vertices. On this reduced graph, the groups of words themselves and the relationships between the groups should be easier to visualize and to interpret. In summary, the idea is to use clustering statistical methods to simplify the original graph.

A first idea would be to use hierarchical cluster analysis or clustering methods based on graph theory as those we used in the last part of Bourgeois et al. [2015]. The problem is that the data we want to deal with is actually too big for those algorithms to be efficient - those ones having running time asymptotically equivalent to n^3 or worse. For this reason, we decided to achieve clustering through the use of Kohonen algorithms whose running time are asymptotically bounded within $O(n^2)$, which remains tractable for a graph of $\sim 10^3$ vertices.

2 Material and Methods

2.1 The material

The corpus (Lamassé [2012], Bourgeois et al. [2015]) includes eight treaties on arithmetical education written in the 15^{th} century, in vernacular language (French). Their purpose was to teach merchants about arithmetics, but also to develop theory and knowledge. One can find a detailed description of their properties and their historical importance in the article mentioned above. Table 1 below describes some elements of the lexicometric characteristics of the corpus and shows its main quantitative imbalance.

Some pre-processing was necessary for these texts. First, the punctuation signs and the *hapax* [2] were removed. After this first step, the data contained 1545 remaining words, including some numbers that are used in the demonstrations or as page numbers.

Second, the co-occurrence matrix was computed. A threshold was used in order to keep the most significant co-occurrences: if $n_{i,j}$ or $n_{j,i} < 5$, both values were set to 0. The value 5 for the threshold was chosen as a compromise between the need for complete information and the need for denoised data. We considered that all co-occurrences smaller than the threshold were random and could be removed. A significantly larger threshold would have lead to a sparser graph,

[2] A *hapax* is a word which appears only once in the corpus.

Manuscripts and **Title**	**Date**	**Author**	**Number of occurrences**	**Words**	**Hapax**
Bibl. nat. Fr. 1339	ca. 1460	anonyme	32077	2335	1229
Bibl. nat. Fr. 2050	ca. 1460	anonyme	39204	1391	544
Cesena Bibl. Malest. S - XXVI - 6, *Traicté de la praticque*	1471?	Mathieu Préhoude?	70023	1540	635
Bibl. nat. Fr. 1346, Commercial appendix of *Triparty en la science des nombres*	1484	Nicolas Chuquet	60814	2256	948
Méd. Nantes 456	ca. 1480-90	anonyme	50649	2252	998
Bibl. nat. Arsenal 2904,*Kadran aux marchans*	1485	Jean Certain	33238	1680	714
Bib. St. Genv. 3143	1471	Jean Adam	16986	1686	895
Bibl. nat. Fr . Nv. Acq. 10259	ca. 1500	anonyme	25407	1597	730

Table 1. Corpus of texts and main lexicometric features

easier to analyze and cluster but suffering from loss of information. Table 2 contains a small sample of the co-occurrence matrix. One should note that it is not a symmetric matrix.

	centre center	*cercle* circle	*chacune* each	*diviser* divide	*endroit* on right	*part* piece	*partie* part	*raisons* calculations, problems
centre	0	23	0	0	0	11	0	0
cercle	20	0	0	0	0	11	0	0
chacune	0	0	0	5	0	0	0	0
endroit	0	0	0	12	0	0	0	0
partie	0	0	0	0	0	0	0	5
raisons	0	0	0	0	0	0	7	0

Table 2. Co-occurrences of eight words extracted from the co-occurrence matrix

Third, the co-occurrence matrix was used to build a weighted and directed graph, where the words represent the vertices and the non-negative co-occurrences represent the weights of the directed edges. At a closer look, the resulting graph contained a large connected component with 1517 words, twelve separate couples of words and one set with four words (all disconnected from each other). The rest of the paper focuses on the largest connected component.

2.2 Robust KORRESP on the co-occurrence matrix

In this first approach, the links between the words in the data set are considered as directed and each word appears both as pole and co-occurrent. Since each word has a double meaning, we shall interpret the co-occurrence matrix C as a contingency table crossing the poles (rows) with their co-occurrents (columns). Factorial Correspondence Analysis (FCA) is in this case the most used and apparently adapted technique. However, due to the limitations of FCA (in terms of projection quality, for example), we propose to use a variant of the SOM algorithm which deals with contingency tables (see Oja and Kaski [1999] for

other applications of SOM to text mining). See Cottrell et al. [1998], Bourgeois et al. [2015] for a definition of this variant of SOM, called KORRESP.

After the convergence of the training step, rows and columns are simultaneously classified and projected on the map. In our example, one shall be able to see proximities between poles, between co-occurrents, between poles and co-occurrents. The goal of Korresp is similar to FCA but its main advantage is that it is not necessary to examine several projection planes : the whole information can be read on the map.

However, Korresp has the drawback of being a stochastic algorithm, and apparent contradictions between several runs can be troublesome. Nevertheless, this drawback can be used to improve the interpretation and the analysis of relations between the studied words. Our hypothesis is that repetitive use of Korresp may allow to identify pairs of words that are strongly attracted/repulsed and fickle pairs.

More precisely, if we consider several runs (at least 50) of the SOM algorithm, for a given size of the map and for a given data set, we observe that most of the pairs are almost always neighbor or not neighbor. But there are also pairs whose associations look random. These pairs are called *fickle* pairs. This question was addressed by de Bodt et al. [2002] in a bootstrap frame and used for text mining in Bourgeois et al. [2015].

After having identified the fickle pairs, we define the fickle words as being those which belong to an important number of fickle pairs. The most fickle words are then removed and the remaining words are then projected on one of the trained maps, hereafter "reference map". In this way, we obtain a map which displays only the robust neighbor relationships.

As each cluster of Korresp is represented by its code-vector, the map provides a new graph where the vertices are the code-vectors. This representation considerably decreases the number of vertices and the complexity of the original graph. Moreover, each code-vector represents a set of words which are very close. Furthermore, hierarchical classification (HAC) may be applied to the code-vectors of the map reducing thus the number of final clusters.

2.3 Robust relational SOM on the co-occurrence matrix

The second algorithm used for exploring the co-occurrence matrix is based on the online version of relational SOM, RSOM (Olteanu and Villa-Vialaneix [2015]). Relational SOM is a generalization of the original SOM algorithm for numerical data. It only requires as input a distance or a dissimilarity matrix between the data. Hence, this method may be applied to any complex data (time series, graphs, texts, etc...) as long as a dissimilarity can be computed.

One of the underlying hypothesis of RSOM is the symmetry of the dissimilarity matrix. The co-occurrence matrix C resuming the texts is not symmetric by construction. Moreover, it translates the strength of the link between the words and hence measures the similarity between them. Matrix C was thus transformed into a symmetric dissimilarity. By doing this, the direct dependency between poles and co-occurrents was lost. However, we shall see in the next sections that

these operations allowed to highlight other aspects and properties of the data set and that the two algorithms are complementary.

Let us briefly explain how the dissimilarity matrix D was computed. First, a new symmetric matrix R was computed from C, where each entry $r_{i,j}$ is

$$r_{i,j} = \frac{1}{2}(n_{i,j} + n_{j,i}).$$

Second, the matrix R was used for building an undirected weighted graph. The vertices of the graph are the words in the texts, regardless of whether they are poles or co-occurrents. The weights, which will later on be used as transition costs, were computed as the inverses of the elements of R. For example, the weight of the edge linking vertex i to vertex j is

$$\omega_{i,j} = 1/r_{i,j}.$$

Third, the dissimilarity matrix D between the vertices is calculated using the shortest-path distance on the weighted graph. The dissimilarity between vertices i and j is then :

$$d_{i,j} = \min_{u_1,u_2 \ldots u_p | u_1=i, u_p=j} \sum_{k=1}^{p-1} \omega_{u_k,u_{k+1}}$$

In order to compare the results of the two approaches (directed links between words versus symmetric links between words), a robust version of RSOM was performed. Similarly to Korresp, RSOM was run with fifty different initializations. The fickle pairs of words and the most fickle words were computed and removed. Then, the remaining words were projected on one of the fifty maps, considered as "reference map". Furthermore, hierarchical classification (HAC) was applied to the code-vectors of the map reducing thus the number of final clusters.

3 Results

Both methods were performed on the selected data, corresponding to the largest connected component of the original graph (1517 words). For each of the robust versions, fifty different initializations of squared grids with 15×15 units were trained. Since both SOM algorithms, Korresp and RSOM, are stochastic or online implementations, the number of iterations was fixed at five times the number of input samples (15 000 iterations for each Korresp map and 7 500 iterations for each RSOM map). The main results and some insights on the specific analysis of the texts are listed below.

3.1 Robust Korresp

The Korresp algorithm performed in its robust version provided the list of the fickle words, ordered in descending order. According to the histogram in Figure 2,

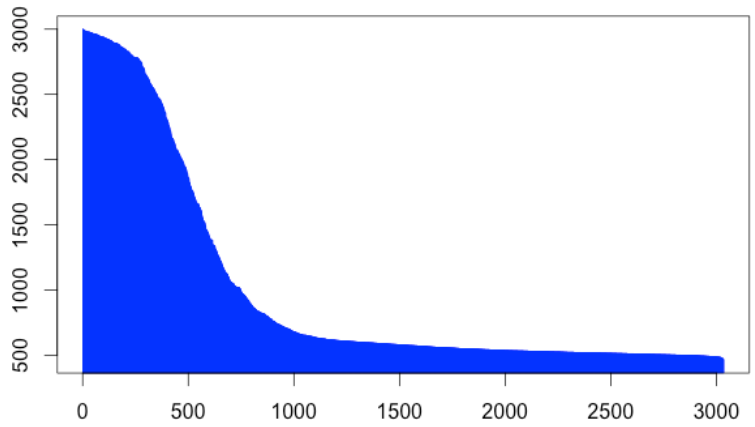

Fig. 2. Robust Korresp : fickle words with the number of their fickle pairs

*partement**(484)	*quant**(484)	*simple**(484)	*demander*(484)	*per*(484)
*excepte**(483)	*toutes**(483)	*on*(483)	*r3**(482)	*mon*(482)
*sommes**(481)	*tes**(481)	*value**(480)	*dont*(480)	*dire**(479)
loing(478)	121(477)	*hommes**(476)	*moien*(474)	105*(472)
*partiteur**(472)	*appartenir*(472)	148385(471)	*mais**(468)	*tenir**(464)

Table 3. Robust Korresp : the twenty-five least fickle words, or the most stable ones. A star means it is the 'pole' form that is fickle.

the most fickle words were removed from the analysis (around 950 words). The least fickle words with regard to the Korresp algorithm are listed in Table 3 below.

The reference Korresp map, containing the remaining least fickle words, and combined with a hierarchical clustering in eleven superclasses is represented in Figure 3. The smoothed distances between the code-vectors are available in Figure 4. According to these two figures, clusters are generally close from one another, except for one region in the large green class, which may suggest the existence of two sub-classes within it.

Cluster 224	Cluster 225
*prix**	147
	surpris
	*juste**

Table 4. Robust Korresp : the contents of super-class 11, composed of clusters 224 and 225

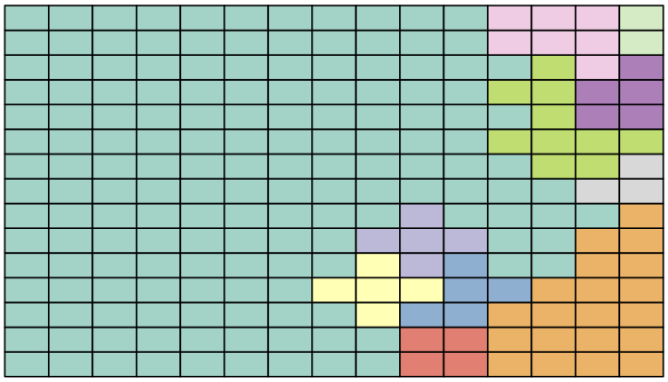

Fig. 3. The robust Korresp. The colors are the super-classes which are determined through a HCA.

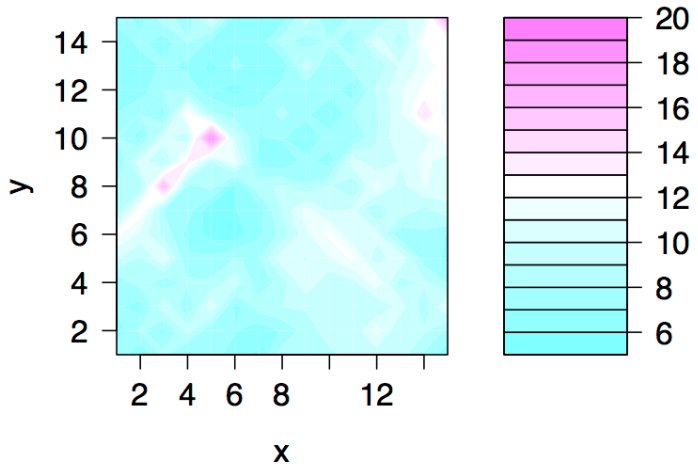

Fig. 4. Robust Korresp : smoothed distances between prototypes

The clusters of the reference Korresp map and the related super-classes are not homogeneous in terms of size and content. For example, let us study the eleventh super-class, colored in light green and situated in the upper-right corner of the map. It is composed of two clusters only, 224 and 225 (the grid is indexed starting from the lower-left corner ; the cluster in the lower-left corner is 1, the cluster in the upper-left corner is 15, etc...). The contents of this super-class are displayed in Table 4.

The clustering and its projection on the map can be further improved. In order to do so, we built a reduced weighted and undirected graph whose vertices

are the clusters of SOM and whose edge-weights are computed as follows

$$A_{I,J} = A_{J,I} = \sum_{i \in I, j \in J} (n_{i,j} + n_{j,i}),$$

where I and J design two clusters, $i \in I$ indexes the words in cluster I and $j \in J$ indexes the words in cluster J. The thickness of the line which joins I and J is proportional to the value of this weight. Hence, the thicker is the edge, the stronger is the link between the clusters. The reduced graph is represented in Figure 5. Since the data used for training the maps was drawn from the largest connected component of the original graph, the reduced projected graph is very connected. Since our goal was to improve the visualisation of the original data by reducing it, two different thresholds were applied to the edges and only the most representative ones were plotted.

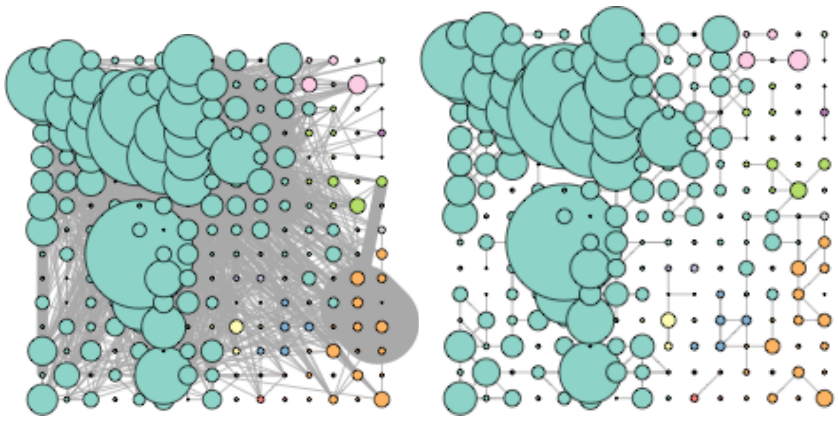

Fig. 5. Robust Korresp : projected graph. Left : only the edges with weights larger than the mean weight value are drawn. Right : only the edges of the four nearest neighbors - left, right, below, above - are drawn. The radius of the clusters is proportional to their size; the width of the edges is proportional to their weight

According to Figure 5, the reference map is composed of a large super-class which contains most of the words while some very specific super-classes emerge very clearly. Since the Korresp algorithm performs on the co-occurrence matrix, it allows to understand the status of words better, poles or co-occurrents.

The first super-class, dark green on the map, is the largest and contains 1871 terms of the 2077 which were clustered. For this large class, it is interesting to zoom further and look within each cluster. For example, cluster 1 deals with the designation of numbers, cluster 11 concerns the computation of interest rates, while cluster 30 concerns ratios of numbers.

The rest of the super-classes contain much fewer terms and are very specific. For example, super-class 2 (light yellow) which is composed of ten poles and

three co-occurrents. The three co-occurents (*vous, gectons, mer*) are specific to abacus computation. Super-class 4 (dark red) is very small, three poles and three co-occurrents. This class is specific to the proof and the verification of results (*vraye, certaine, preuve, prouver, bien, venue*).

We also notice here the effect of the absence of lemmatization, since some close words appear in the same cluster (*compagnon, compaignon*).

3.2 Robust relational SOM

The histogram of the fickle words provided by the robust version of RSOM and ordered in descending order is plotted in Figure 6. We notice that in this case, most of the words have a high index of fickleness. This seems to be mainly due to the fact that the syntax of the texts was suppressed when symmetrizing the co-occurrence matrix.

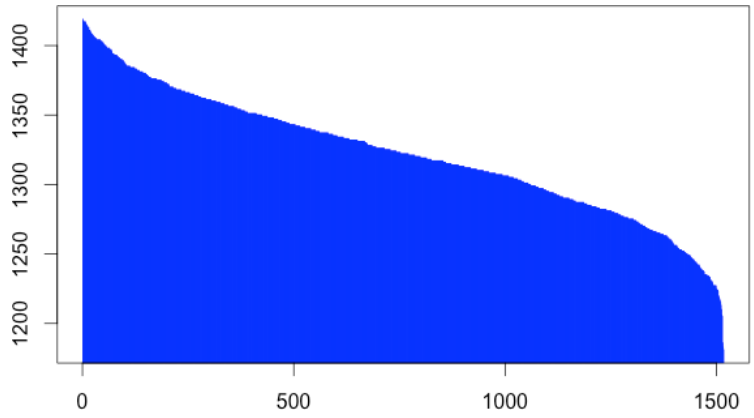

Fig. 6. Robust RSOM : fickle words with the number of their fickle pairs

The least fickle words with regard to the RSOM algorithm are listed in Table 5 below. The first remark one can make is that the most fickle words in this case are generally numbers (Roman or Arabic) and pronouns. The least fickle words are generally nouns and verbs. Hence, it appears that the use of RSOM and the symmetrization of the co-occurrence matrix is more specifically adapted for highlighting lexical aspects, while the directed co-occurrence matrix used together with Korresp seemed to be more specific for the semantics of the texts. Before building the reference RSOM map, around fickle 200 words were removed.

The reference RSOM map, combined with a hierarchical clustering in eight superclasses is represented in Figure 7. The smoothed distances between the code-vectors are available in Figure 8. According to these two figures, several large distances colored in dark pink suggest the existence of several super-classes. The composition of the super-classes in terms of SOM clusters appears more

condition(1227)	*cubbe*(1227)	*gnomon*(1227)	*marcz*(1227)	*mon*(1227)
precedente(1227)	*racine*(1227)	*sur*(1225)	*appellee*(1224)	*douzaine*(1224)
nous(1224)	*gaigneroye*(1223)	*superparticuliere*(1220)	*droit*(1219)	*mars*(1218)
layne(1217)	*precedentes*(1216)	*secondement*(1215)	*draps*(1213)	*noter*(1210)
car(1206)	*disain*(1205)	*gangne*(1186)	*nommer*(1183)	*notable*(1181)

Table 5. Robust RSOM : the twenty-five least fickle words, or the most stable ones

homogeneous than previously and one may expect well-balanced super-classes. When studying the contents of the super-classes, the number of elements in each of them is between 100 and 280.

Similarly to robust Korresp, a reduced graph with the SOM clusters represented as vertices is build and projected on the map, as illustrated in Figure 9. We shall remark here that, since the lexical aspect is more powerful here than the semantics, clusters are almost all linked together, even after thresholding the edges to plot.

Let us now take a closer look at some of the super-classes. The first super-class (dark green, lower-left part of the map) appears to contain the lexical ensembles which structure the text in a mathematical problem. For instance, clusters 1 16 are clearly containing the words used in the statement of a mathematical problem. The second super-class (orange, left) groups mathematical terms such as sums, numbers and quantities. The third super-class (blue, upper-left) contains almost exclusively Arabic numbers. In the fourth super-class, we find words which design numbers (*digit, articles, composes, mixte*) and the specific technique of abacus calculation (*gectons, gect*).

4 Conclusions and perspectives

The detailed description of the projection results on the two reference maps was not addressed here since, on the one hand, the number of pages is limited and, on the other hand, the specificities of the corpus of documents would only passionate the specialists of the discipline. However, let us remark that Korresp seems more likely to be adapted to semantic issues and that it appears to isolate and highlight very specific groups of words. At the same time, RSOM seems more likely to provide lexical insights on the studied texts. These characteristics are due to the fact that Korresp is trained on the original co-occurrence matrix C, which is directed, while RSOM is trained on a symmetric dissimilarity matrix computed from C.

Although each of them has its specificities, both methods, Korresp and RSOM, may be viewed as useful tools for simplifying, reducing and visualizing complex graphs, of high dimensionality. However, one of the current drawbacks of the two algorithms is the computational time, especially for RSOM where the code-vectors are supposed to be linear combinations of the original input data. The latter hypothesis could be weakened, especially as the co-occurrence

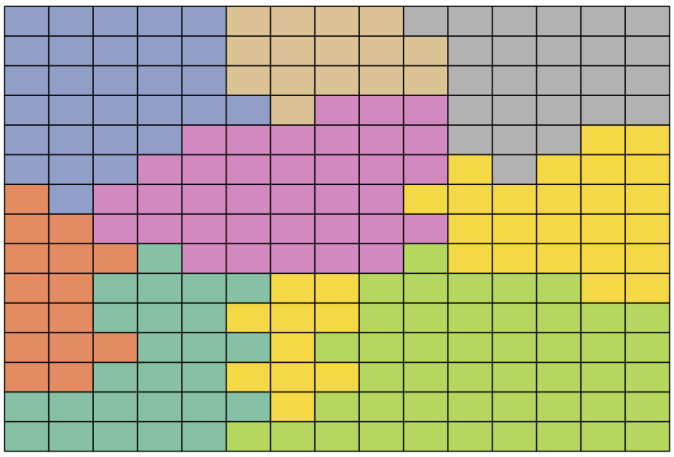

Fig. 7. The robust relational SOM. The colors are the super-classes which are determined through a HCA.

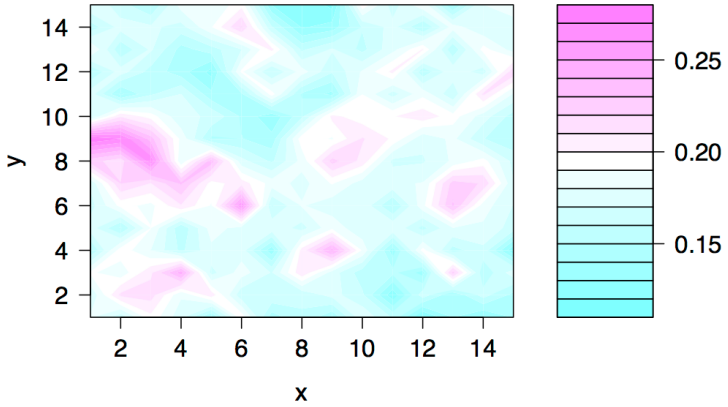

Fig. 8. Robust relational SOM : smooth distances between prototypes

matrices are generally very sparse. Hence, the next step of our research is to adapt the two algorithms to very large and sparse matrices and test them with big text-mining data.

References

Jean-Pierre Benzecri. *Correspondence Analysis Handbook.* Marcel Dekker, New-York, 1992.

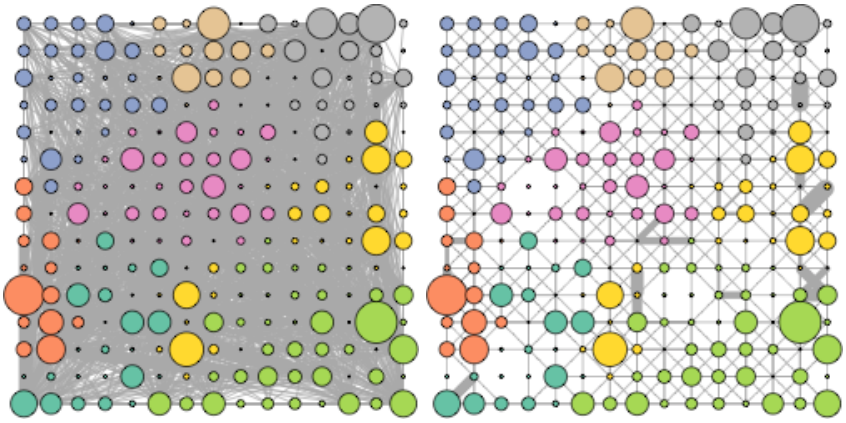

Fig. 9. Robust RSOM : projected graph. Left : only the edges with weights larger than the 3rd quartile are drawn. Right : only the edges of the four nearest neighbors - left, right, below, above - are drawn. The radius of the clusters is proportional to their size; the width of the edges is proportional to their weight

Davi Blei. Probabilistic topic models. *Communications of the ACM*, 55(4): 77–84, 2012.

Nicolas Bourgeois, Marie Cottrell, Benjamin Deruelle, Stéphane Lamassé, and Patrick Letrémy. How to improve robustness in kohonen maps and display additional information in factorial analysis: Application to text mining. *Neurocomputing*, 147:120–135, 2015.

Marie Cottrell, Jean-Claude Fort, and Gilles Pagès. Theoretical aspects of the SOM algorithm. *Neurocomputing*, 21:119–138, 1998.

Eric de Bodt, Marie Cottrell, and Michel Verleysen. Statistical tools to assess the reliability of self-organizing maps. *Neural Networks*, 15, 8-9:967–978, 2002.

Stéphane Lamassé. Les traités d'arithmétique médiévale et la constitution d'une langue de spécialité. In Joëlle Ducos, editor, *Sciences et langues au Moyen Âge, Actes de l'Atelier franco-allemand, Paris, 27-30 janvier 2009*, pages 66–104. Universitätesverlag, Heidelberg, 2012.

Ludovic Lebart, Alain Morineau, and Kenneth Warwicki. *Multivariate Descriptive Statistical Analysis Correspondence Analysis and Related Techniques for Large Matrices*. Wiley, New-York, 1984.

William Martinez. Au-delà de la cooccurrence binaire...poly-cooccurrence et trames de cooccurrence. *Corpus*, 11:191–216, 2012.

William Martinez and André Salem. *Contribution à une méthodologie de l'analyse des cooccurrences lexicales multiples dans les corpus textuels*. Thèse doctorat, 2003.

Erkki Oja and Samuel Kaski. *Kohonen Maps*. Elsevier, 1999.

Madalina Olteanu and Nathalie Villa-Vialaneix. On-line relational and multiple relational som. *Neurocomputing*, 147:15–30, 2015.

Mike Titterington, Adrian Smith, and Udi Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.