Visualiser les textes et les mots : approches numériques, approches par les graphes

Alain Lelu

Chapitre 5

Visualiser les textes et les mots :

approches numériques, approches par les graphes

5.1. Introduction

On rencontre sur le Web de plus en plus de représentations visuelles et graphiques de grandes masses de textes et de grandes quantités de mots, que ceux-ci soient issus de corpus tels que les pages Web ou les bases de résumés d'articles scientifiques, ou de ces corpus particuliers qui définissent un mot par un texte, que sont les encyclopédies, dictionnaires et dictionnaires analogiques. Ces représentations synthétiques permettent à l'usager de se faire une idée d'ensemble de tout ou partie d'un corpus qui l'intéresse, centré par exemple autour d'une requête explicite par mots, ou implicite par désignation d'un objet d'intérêt. On peut citer ainsi les représentations interactives qu'on trouve sur le site Touchgraph.com qui permettent de naviguer entre autres dans le fonds d'Amazon.com[1], de façon convaincante, ou dans les pages résultant d'une requête Google à partir de leurs liens[2] ; les sites http://prox.irit.fr (voir figure 5.1) et http://elsap1.unicaen.fr/cgi-bin/cherches.cgi donnent accès aux proximités sémantiques entre 49 000 verbes,

---

Chapitre rédigé par Alain LELU

[1] www.touchgraph.com/TGAmazonBrowser.html
[2] www.touchgraph.com/TGGoogleBrowser.html

noms, et adjectifs de la langue française. Beaucoup de secteurs des sciences humaines, qu'elles soient universitaires ou appliquées comme les études de marché, utilisent depuis bon nombre d'années ces représentations qu'ils ont contribué à initier et perfectionner.

La création de ces représentations visuelles et interactives se situe au confluent de trois domaines : celui à caractère mathématique des méthodes dites de réduction de dimensions des données, celui des techniques informatiques de visualisation graphique interactives, (de plus en plus dynamiques et tridimensionnelles, voir figure 5.1), et celui à caractère largement artistique et ergonomique du *design* des interfaces. C'est sur le premier domaine que se concentre le présent chapitre.

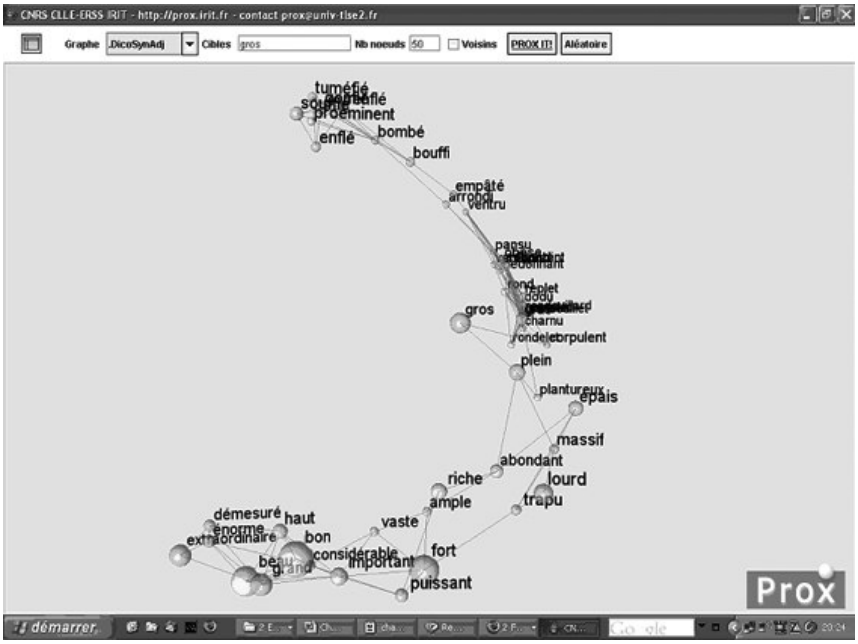

Figure 5.1. *Représentation de l'espace sémantique de l'adjectif* gros *en 3D interactive sur le site ILAN Prox.*

Notre objectif est d'offrir un panorama des méthodes qui permettent d'engendrer de telles représentations. Ce domaine est à la fois ancien – les premières cartes factorielles de corpus indexés remontent à notre connaissance aux travaux de J.P.

Benzécri à la fin des années 1960 [BEN 81] – et objet d'un intérêt constamment renouvelé depuis.

Une telle dynamique ne semble pas près de s'arrêter, car de nombreuses difficultés subsistent, comme nous le verrons. Un consensus ne s'est pas encore dégagé sur certaines formes plutôt que d'autres. Pour prendre la métaphore de l'histoire de la bicyclette, nous en sommes encore à l'époque des draisines, des « grands bis », des tentatives de mettre le cycliste à l'intérieur de la roue, etc., et l'interaction entre les usages et les possibilités techniques – mathématiques et informatiques dans notre cas – n'a pas encore stabilisé de forme dominante[3], comme a pu l'être au début du 20e siècle le vélo standard, coexistant avec un petit nombre de formes marginales, comme la trottinette ou le monocycle de cirque.

Pour revenir à la représentation visuelle des textes, la taille du problème posé – toute langue comporte des dizaines de milliers de mots, des centaines de milliers d'expressions composées, les pages Web se comptent par milliards, les résumés d'articles scientifiques par millions, même si l'on ne s'intéresse en pratique qu'à de « petits » sous-ensembles de quelques milliers ou dizaines de milliers d'unités - tend à disqualifier les approches symboliques, sujettes à l'explosion combinatoire, au profit des approches numériques. Ce qui signifie que toute collection d'unités textuelles – textes, mots, pages Web, … - doit être réduite à une collection de vecteurs, en d'autres termes à un tableau (descripteurs × objets décrits) de valeurs numériques (généralement avec une immense majorité de zéros ; on parle alors de vecteurs et matrice *creux*). Ce qui constitue un premier problème de fond, dont nous présenterons les principales solutions, le plus souvent issues du domaine du Traitement Automatique des Langues Naturelles, et discuterons les limites, ainsi que quelques travaux actuels pour repousser ces limites.

Le deuxième problème de base est de passer de ces vecteurs définis dans un espace fortement multidimensionnel (par exemple, celui des mots utilisés dans un corpus) aux deux ou trois dimensions des représentations graphiques accessibles à l'œil et à l'esprit humain. Dans ce cas on dispose d'un socle de méthodes bien établies dans d'autres domaines scientifiques, certaines depuis longtemps (l'analyse factorielle des psychologues remonte aux années 1900…), qui doivent alors être transposées, modifiées, « passées à l'échelle » dans notre domaine d'application. Des modes de représentation et de navigation dans des contenus textuels peuvent alors voir le jour, et se confronter aux usages des milieux scientifiques demandeurs

[3] On peut cependant constater que sur les sites de type « Web 2.0 » utilisant les interactions sociales collaboratives (*social bookmarking*), la forme de cartographie « nuage de mots-clés » (*tag cloud*), rudimentaire à nos yeux puisqu'elle consiste à écrire une liste alphabétique de mots couvrant l'ensemble d'un cadre avec des grosseur de caractères reflétant le nombre d'utilisations par les internautes (ou par l'auteur), semble se généraliser (voir figure 5.2).

et du grand public, donnant le jour à de nouvelles exigences. En parallèle, des méthodes nouvelles issues de problématiques d'ingénierie de processus, de traitement du signal ou de l'image, des études sur les graphes, de la bioinformatique, continuent à fertiliser le domaine de l'analyse de données textuelles, aboutissant au foisonnement actuel.

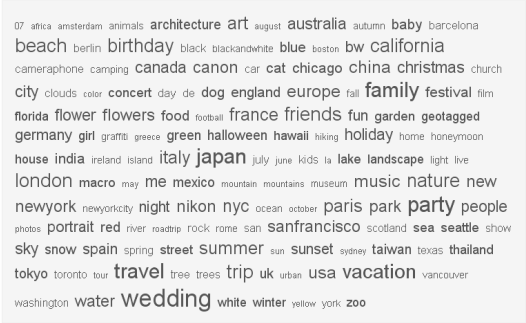

Figure 5.2 : *nuage de mots-clés (*tags*) les plus populaires attribués aux photos du site FlickR.*

Impossible d'être exhaustif face aux centaines, voire milliers de travaux par an qui s'attaquent aux multiples facettes de la réduction de dimensions dans le contexte d'application aux faits et produits de la langue. Nous tenterons seulement d'en dégager les familles principales débouchant sur des représentations visuelles, représentations que les progrès de l'informatique et la créativité des designers d'interface rendent de plus en plus attrayantes pour l'œil et, on l'espère, explicites pour l'esprit humain.

5.2. Des textes aux vecteurs

Loin du monde idéal des grammairiens, un texte du monde réel est un objet très structuré, mais aussi imparfait, comme beaucoup d'entités biologiques, avec des ratés, des accidents, des redondances, des branches mortes… Le réduire sans trop y perdre à un vecteur, c'est-à-dire à un vrac de valeurs de descripteurs, loin de toute notion de séquence et de structure, n'est pas évident, et dépend fortement de l'usage que l'on veut faire de la représentation visée. Sauf pour de rares publics de spécialistes de la langue, le but des méthodes de visualisation est de traduire de façon condensée « de quoi » parlent les textes. La façon dont ils en parlent, les assertions, raisonnements, opinions qu'ils déploient à leur propos ne sont pas les

préoccupations prioritaires d'un utilisateur d'interface vers une base de textes ou d'objets à descriptions textuelles[4].

Ceci amène à éliminer un maximum de mots à rôles syntaxique, rhétorique ou narratif. Mais repérer le rôle des mots et, quand elles sont figées, des combinaisons de mots, est un problème en soi, sur lequel les méthodes du Traitement Automatique de Langues Naturelles (TALN) et de la linguistique de corpus ont apporté des solutions, imparfaites, mais suffisantes pour le seul objectif de visualisation des contenus sémantiques d'une collection de textes. Nous verrons à la fin de cette section comment certains problèmes posés par la réduction de chaque morceau de texte à un « sac de mots » font l'objet de recherches et de tentatives de solution.

5.2.1. Un cas simple, l'indexation manuelle et les dictionnaires analogiques

Dans un certain nombre d'applications, le problème ne se pose pas ou peu : quand chaque objet d'une collection est décrit par un nombre indéterminé d'étiquettes placées manuellement, le passage aux vecteurs est trivial. C'est le cas des bases documentaires indexées manuellement, c'est aussi le cas pour les services Web 2.0 à forte interactivité personnelle ; ici l'indexation manuelle prend le nom de *social tagging*, dans laquelle l'attribution d'étiquettes (*tags*) aux films, musiques, personnes, etc. fait l'objet d'enjeux personnels forts de la part de l'usager, à savoir définir une communauté d'intérêts – et par là se définir, faire des rencontres… Ces étiquettes forment ainsi pour chaque entité de la collection une description textuelle rudimentaire et minimale.

Ce type d'indexation peut être généralisé : à chaque page Web sont associés des liens non ambigus vers d'autres pages - cette indexation par les liens est à la base de l'efficacité de Google. A chaque enregistrement d'article scientifique sont associées des citations, entrantes ou sortantes, d'ambiguïté variable selon la base documentaire qui les édite. La base Web of Science[5] doit ainsi une grande partie de son caractère incontournable pour les études scientométriques au codage manuel des listes de références bibliographiques qu'elle pratique depuis sa création. Bien qu'imparfait, il

[4] Le courant informatique dit d'extraction de connaissance à partir des données (*Knowledge Discovery*) vise à traduire les textes informels en connaissance structurée et à assurer un accès « intelligent » à cette structure en introduisant des raisonnements automatiques dans le processus de réponse aux requêtes de l'utilisateur. Pour l'instant le passage de ce projet, qui s'appuie sur les formalismes et les normes du Web Sémantique (dit désormais Web 3.0), à l'échelle réelle semble difficile, même pour répondre à des questions spécialisées d'un domaine scientifique lui-même spécialisé (par ex., les interactions entre gènes), du fait du caractère indiscipliné du langage naturel, qui peut cacher autant que montrer, et rendre ambigu autant qu'expliciter…même dans les articles scientifiques.

[5] http://www.scientific.thomson.com/products/wos/

surpasse pour l'instant les recodages automatiques pratiqués par les bases de citations *open access* comme CiteBase[6] et GoogleScholar[7].

Tout dictionnaire analogique se compose, pour chaque mot d'entrée, d'une liste de mots analogues. Il est facile de le traduire en un ensemble de vecteurs binaires constituant une grande matrice dite d'adjacence, a priori carrée. L'informatique permet de vérifier si des relations d'analogie, caractéristique considérée comme réflexive (si *épanoui* est un analogue de *jovial*, alors *jovial* est analogue d'*épanoui*) n'ont pas échappé à la vigilance des auteurs – ce n'est bien sûr pas le cas pour les dictionnaires établis avant l'arrivée de l'informatique... La base WordNet[8] comporte des ensembles de synonymes (SynSets) et a été utilisée comme source de synthèses visuelles d'associations entre mots par de nombreux auteurs.

5.2.2. Des textes bruts aux vecteurs de termes : indexation et TALN

En introduction de la section, nous avons vu que l'objectif à réaliser était, pour parler comme la Grammaire de Port-Royal, d'éliminer le « rhème » pour ne garder que le « sème ». Quelques grandes familles de méthodes permettent d'extraire des textes bruts les descripteurs pertinents de leur contenu :

– la méthode informatiquement la plus simple consiste à considérer toute chaîne de caractères entourée de séparateurs (espace, point, …) comme un descripteur ; on la nomme indexation en texte intégral en français, *full-text indexing* en anglais, et elle reste très utilisée dans beaucoup de moteurs de recherche et de logiciels documentaires, sans parler de sites Web 2.0. Son avantage est qu'elle se prête idéalement aux requêtes ponctuelles fines pour retrouver par exemple un nom d'individu ou un fragment précis de texte. Son inconvénient, indépendamment des questions de volume de stockage d'information, est qu'elle ne filtre rien. On peut lui adjoindre des perfectionnements de deux ordres : 1) des unifications de chaînes de caractères par simple troncature (*stemming* en anglais, par ex. suppression des *s* finaux) ; 2) des anti-dictionnaires (*stop-lists*) ou listes de chaînes à éliminer. Ce qui constitue un premier filtrage, rudimentaire et bruité, pour éliminer au moins la plupart des mots grammaticaux – ceci n'étant opératoire que sur des corpus monolingues, ou des systèmes incorporant la reconnaissance de la langue ;

– l'analyse morpho-syntaxique attribue à chaque chaîne de caractères du texte un lemme (forme normalisée) ainsi qu'une catégorie et un rôle syntaxique. On trouve dans le domaine public de tels analyseurs, à apprentissage probabiliste comme Tree Tagger[9], ou à automates d'états finis comme NOOJ[10]. Il est alors possible de filtrer

[6] http://www.citebase.org/
[7] http://scholar.google.fr/
[8] http://wordnet.princeton.edu/
[9] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

globalement par exemple l'ensemble des verbes, ou l'ensemble des adjectifs, en tant que participant plus du rhème que du sème. Mais ceci est discutable dans les contextes scientifiques, techniques ou juridiques spécialisés, où beaucoup de verbes et adjectifs peuvent être nominalisés, et donc porter des contenus précis. A l'inverse, beaucoup de substantifs peuvent participer du rhème, comme *développement*, *résultat*, … dans le domaine scientifique, et ne produire que du bruit ;

– deux éléments de contenu essentiels peuvent être dégagés de façon relativement indépendante de la structure morpho-syntaxique :

- extraire les *entités nommées* consiste à regrouper, unifier les variantes multiples d'un nom de personne, de ville, d'entreprise, de produit, d'appareil, de substance… Il s'agit d'un problème difficile, ces variantes pouvant être ou non proches lexicalement. Nos travaux ont abouti à une procédure semi-automatique en ce domaine, à savoir doter l'indexeur d'outils pour lui montrer à la fois les termes les plus proches lexicalement d'une chaîne donnée [LEL 00a] et les termes proches sémantiquement, par similarité vectorielle, d'un terme donné [LEL 00b]. D'autres travaux utilisent des approches d'apprentissage automatique, ou d'automates d'états finis. Pour une synthèse de l'état de l'art, voir [NAD 06] ;

- les expressions composées sont des séquences de mots figées par l'usage et portant un sens précis, non réductible à celui de leurs composants (*bases de données*, *foreign minister*, …). Les détecter constitue un enjeu capital pour la représentation du contenu dans les domaines spécialisés, scientifiques ou autres. Elles posent trois problèmes redoutables :

- leur degré de figement est variable (*pomme de terre* ne pose pas de problème, mais faut-il accepter *spatial direction* ou *microscopic level* ?),

- elles peuvent comporter des enchâssements multiples (dans *probabilistic model retrieval*, faut il voir *probabilistic model*, *probabilistic retrieval*, *probabilistic model retrieval* ?),

- beaucoup d'entre elles participent du rhème, et sont à éliminer (*alternative method*, *interesting results*, …). Les travaux dans ce domaine sont nombreux et souvent basés sur des principes d'analyse syntaxique superficielle à partir du repérage des groupes nominaux par des marqueurs de frontières (*chunking*), mais n'ont pas abouti, à notre connaissance, à des solutions automatiques satisfaisantes selon ces trois critères.

5.2.3. Limites de ces approches et directions de recherche

Concernant le dernier point mentionné ci-dessus, signalons qu'après avoir proposé des réponses empiriques semi-automatiques à ces problèmes, implantées dans l'environnement de visualisation d'information NeuroNav issu de nos

[10] http://www.nooj4nlp.net/

recherches (voir figure 3), nos travaux actuels portent sur des procédures de validation statistique des combinaisons de 2 mots et plus au sein d'un corpus, par des tests de randomisation, indépendants des distributions sous-jacentes, et adaptés au cas d'un grand nombre de descripteurs [CAD 07], à la différence des tests statistiques classiques.

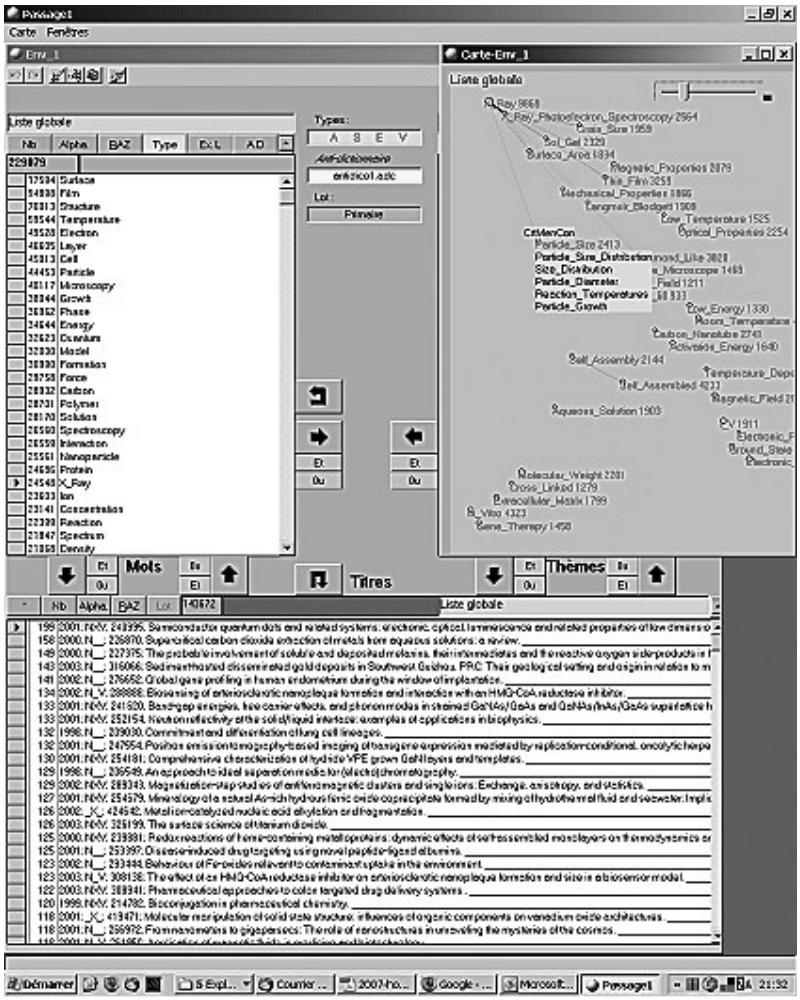

Figure 5.3 : *environnement de contrôle de vocabulaire et cartographie textuelle NeuroNav®*

De façon générale, compte tenu du caractère majoritairement polysémique des mots de la langue (même celle des scientifiques !), une représentation du contenu d'un texte par un sac de mots peut être sémantiquement ambiguë et bruitée. C'est pourquoi certains auteurs tentent d'enrichir sémantiquement la représentation d'un texte, tout en restant dans le cadre du modèle vectoriel. Ainsi les auteurs [RAJ 00] multiplient chaque vecteur-texte (fréquences de mots) par une matrice de co-occurrences *filtrées* ; on peut par exemple compter les co-occurrences sous la contrainte syntaxique de se produire dans une même partie de phrase. De cette façon, dans « *Le mouton frisé broute tranquillement à côté de la barrière moussue* », les co-occurrences *barrière-moussue* et *mouton-frisé* sont à garder, à la différence de *frisé-barrière*, *mouton-moussue* et *frisé-moussue*. Elles ressortent ainsi parce qu'on a remplacé le profil classique d'occurrence des mots dans un document par la somme des profils de co-occurrence de tous les mots avec les mots d'origine du texte. Ceci revient à remplacer ce qui a été dit explicitement par l'auteur par la somme des contextes des mots qu'il a prononcé – en cas d'ambiguïtés sur plusieurs mots d'origine, un seul contexte sémantique émergera.

Plus fondamental encore : la représentation basique des textes par des vecteurs d'occurrence de mots suppose implicitement que les valeurs zéro, en général très majoritaires, sont issues d'un choix délibéré de l'auteur de ne pas employer tel ou tel mot. Ce n'est bien sûr pas le cas, un mot pouvant avoir été choisi à la place d'un autre pour des raisons de pure contingence ou d'esthétique. Des méthodes issues du domaine nouveau dénommé filtrage collaboratif [HER 04] permettent de considérer ces zéros comme des valeurs manquantes dont on peut tenir compte dans l'analyse, ou qu'il est possible de reconstituer, y compris quand la matrice de données est très creuse.

5.3. Des vecteurs aux représentations 2D et 3D

Une fois un ensemble de textes transformé en un ensemble de vecteurs, un monde de possibilités s'offre pour en tirer une représentation graphique. En effet le formalisme des vecteurs et des matrices (une matrice = un ensemble de vecteurs de mêmes dimensions) ouvre les portes d'un domaine mathématique longuement balisé depuis plus de deux siècles dans les sciences « dures » : l'algèbre linéaire ; domaine à la base de la plupart des applications numériques de l'informatique – traitement d'image, et plus généralement du signal, simulations en tous genres (climat, économie, ingénierie, …), conception assistée par ordinateur, …

Si *linéaire* signifie, dans algèbre linéaire, que les effets sont supposés s'additionner proportionnellement aux causes, on trouve de plus en plus d'extensions non linéaires des méthodes linéaires, dont nous présenterons les principes et verrons l'intérêt plus bas. Loin d'être figés, ces développements

mathématiques et statistiques se poursuivent de plus belle sous l'aiguillon des problèmes nouveaux nés de la numérisation croissante des textes, des sons et des images.

La métaphore spatiale que permettent ces méthodes – représenter un ensemble d'items, par exemple des textes ou des pages Web, par des points dans 2 ou 3 dimensions – prend sa source dans la possibilité qu'on a de mesurer une similarité, ou une distance, entre deux vecteurs. Dès lors il est possible de rassembler les valeurs de similarité entre tous les vecteurs de la collection dans une matrice carrée de similarité, quelle que soit la méthode parmi l'une des nombreuses possibles pour calculer ces similarités (voir figure 5.4).

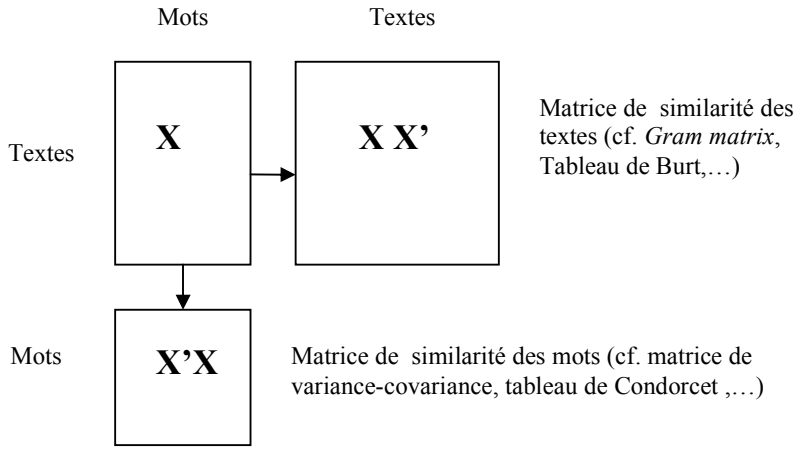

Figure 5.4 : *Tableau de données et matrices de similarité élémentaire entre lignes / entre colonnes*

C'est à partir de là que les trois grandes familles de méthodes utilisables divergent : la famille des analyses factorielles et de leurs nombreuses variantes et extensions utilise explicitement ou implicitement la totalité de l'information de similarité, la famille de la représentation par graphes utilise des seuillages ou transformations sur cette matrice pour ne garder que les liens de similarité les plus forts entre textes[11], la famille « clustering » et facteurs obliques aboutit à une représentation à deux étages, où une carte globale décrit non des textes ou des mots, mais des regroupements homogènes de textes (*clusters*), sur lesquels il est possible de zoomer.

[11] Ou entre mots si on utilise une matrice de similarité des mots.

5.3.1. *La voie factorielle classique*

L'*analyse factorielle* ([SPE 04], [PEA 01]) et l'*analyse en composantes principales* (ACP) [HOT 33]) ont été conçues dans le premier tiers du 20ème siècle, mais sont restées confinées à des applications de taille très limitée, principalement en psychologie, avant l'arrivée de l'informatique dans les années 1950 et 1960. Leur principe est simple :

– chaque vecteur-donnée **x** (individu, ou observation), à I dimensions, autant que de variables (ici, principalement des mots), est exprimé comme une somme pondérée de K (où $K \leq I$) "composantes" $\mathbf{w}_k$, appelées aussi facteurs communs ;
– chaque composante traduit une variable "latente", cachée dans les données ;

selon ses variantes, ce modèle s'écrit :

$\mathbf{x} = \mathbf{W}\mathbf{y} + \mathbf{e}$ (analyse factorielle)[12]

$\mathbf{x} = \mathbf{W}\mathbf{y}$ (cas de l'ACP avec K=I quand **X** est de rang plein)

où **y** est le vecteur des coordonnées factorielles (*factor score*) de l'individu **x**, **W** est la matrice formée par l'ensemble des vecteurs $\mathbf{w}_k$ et **e** un vecteur bruit, spécifique de cet individu.

Ce type de modèle (voir figure 5.5) est décliné sous de nombreuses formes, a eu et continue d'avoir une riche descendance au fur et à mesure que la puissance informatique disponible augmente.

[12] Nous noterons, de façon générale, les scalaires en caractères maigres, les vecteurs en minuscules grasses, les matrices en majuscules grasses et leurs transposées en ajoutant un signe prime.

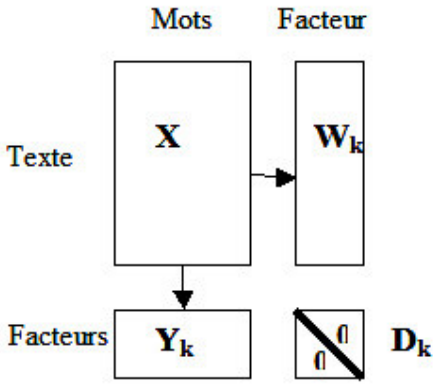

Figure 5.5 : *Matrice de données **X**, matrices W_k et Y_k des k premiers facteurs lignes et colonnes, D_k des valeurs propres*

- L'*analyse en composantes principales* est une méthode devenue standard dans de nombreux domaines scientifiques, où elle porte parfois des noms différents (transformée de Karhunen-Loeve, d'Hotelling...). Le tableau de données **X** comporte I variables centrées et N observations[13], et sa décomposition est

$\mathbf{X} = \mathbf{W}\mathbf{D}^{1/2}\mathbf{Y}$, où **D** est la matrice diagonale des I valeurs propres obtenue à partir de la décomposition spectrale de la matrice de variance-covariance des données (1/N) **X'X** en composantes orthonormales non-corrélées :

$\mathbf{X'X} = \mathbf{Y'DY}$ (formule de reconstitution de la variance-covariance)

$\mathbf{W} = \mathbf{XY'D}^{-1/2}$ (formule de transition entre facteurs-colonnes et facteurs-lignes)

On s'intéresse généralement aux éléments propres des k premiers rangs, qui donnent souvent lieu à des cartes représentant soit les individus, soit les variables, soit les deux, les autres éléments propres étant considérés représenter le "bruit" dans les données :

$\mathbf{X} \approx \mathbf{W}_k \mathbf{D}_k^{1/2} \mathbf{Y}_k$ (reconstitution de rang k des données, optimale au sens des moindres carrés)

[13] On supposera, sans perte de généralité, que **X** est de rang I (quand N>I), ce qui est le plus souvent vrai pour les données d'observation empirique.

Une variante importante en est l'*analyse factorielle des correspondances* [BEN 80] qui utilise la métrique du Chi2 et permet d'analyser les tableaux de contingence de façon « symétrique », avec une représentation simultanée des points-lignes et des points-colonnes dans les plans factoriels. En l'occurrence, un tableau de comptages (Documents × Mots) est un tableau de contingence pour lequel la propriété d'*équivalence distributionnelle* de l'AFC est particulièrement appropriée – il s'agit de la propriété de stabilité de l'analyse vis-à-vis des découpages ou regroupements de documents ou mots, si les profils relatifs décrivant ces documents (ou mots) regroupés ou éclatés se ressemblent. Ce type d'analyse produisant souvent des nuages de points très « tassés » au centre, une heuristique commode bien qu'arbitraire consiste à représenter chaque point non pas par ses coordonnées factorielles, mais par les *rangs* de ses coordonnées sur chaque facteur. (voir figure 5.6)

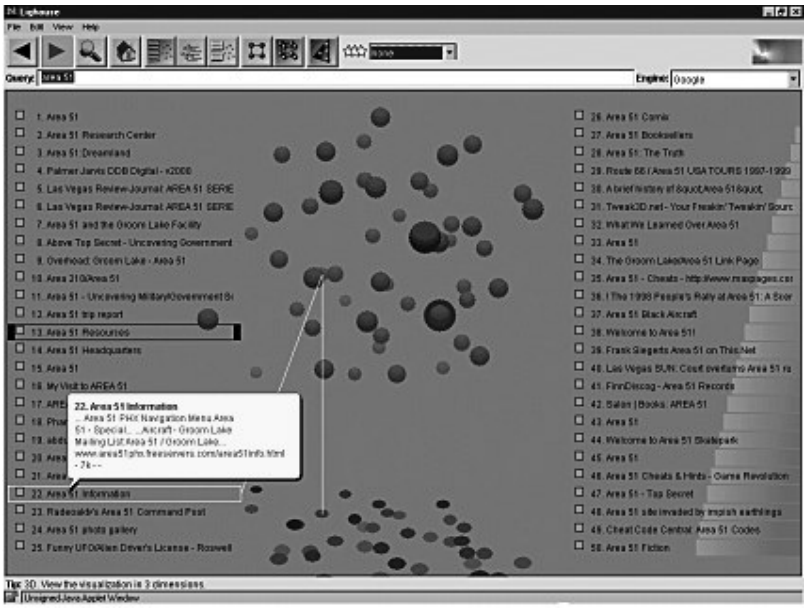

Figure 5.6 : *représentation d'un nuage de points (1 point = 1 site Web) dans 3 facteurs. Source : A. Leuski et cybergeography.org*

5.3.2. *Extensions non linéaires de la voie factorielle classique et dépliage multidimensionnel*

On peut schématiser comme suit le principe des algorithmes classiques de la famille factorielle :

– On converge vers un premier axe qui maximise l'étalement des projections du nuage de points, au moyen de techniques dites de montée en gradient sur une fonction objectif indicatrice de cet étalement, comme la somme des carrés des projections – cette procédure permet d'atteindre un maximum absolu et unique (axe d'allongement principal du nuage).

– On retire des données la « part expliquée par le premier axe » - mathématiquement, on soustrait de chaque vecteur-donnée son vecteur-projection sur l'axe ; le nuage de points se trouve alors confiné dans un hyperplan orthogonal à cet axe.

– On itère ce processus sur les données ainsi transformées, pour obtenir un deuxième axe, éventuellement un troisième, orthogonaux entre eux par construction.

Par rapport à ce processus hiérarchique, l'idée du dépliage multidimensionnel est de trouver une représentation globale en 2D ou 3D : 1) pour laquelle les distances dans cet espace réduit soient le plus proches possibles des distances dans l'espace d'origine, 2) tout en privilégiant généralement l'exactitude des faibles distances, donc des relations de voisinage « vraies », au détriment des fortes distances. Par exemple un « filet » de points plié de façon complexe dans l'espace d'origine (*variété non linéaire*, en termes mathématiques) pourra se trouver déplié de façon claire et lisible dans deux dimensions seulement. Plusieurs méthodes le permettent :

– La plus ancienne, l'échelonnement multidimensionnel (MDS, *Multidimensional Scaling*) [KRU 64], opère par minimisation d'une fonction de perte, par exemple la somme des carrés des différences entre distances d'origine et distances dans 2D (ou 3D) pour les N points, pondérée par une fonction monotone des distances d'origine. Selon son allure, cette fonction privilégiera plus ou moins les faibles distances au détriment des grandes. Une technique d'optimisation possible parmi d'autres est le recuit simulé, où l'on part d'un semis au hasard de N points dans 2D ou 3D, qu'on cherche à améliorer par de petits déplacements au hasard : si un déplacement élémentaire fait baisser la fonction de perte, il est retenu ; un paramètre dit de température permettra de retenir aussi des déplacements non optimaux, selon une certaine probabilité, de façon à éviter de rester coincé dans des minima locaux loin de l'optimum global, qu'on sait inatteignable en pratique. Au fil des très nombreuses itérations nécessaires, la température sera abaissée progressivement, et la convergence s'opérera vers une solution satisfaisante, proche de l'optimum absolu.

– D'autres méthodes étendent de façon non linéaire les méthodes d'analyse factorielles classiques : l'*Analyse Discriminante Intrinsèque* [BUR 91] réalise l'ACP d'une matrice de variance/covariance locale, dérivée de la matrice des similarités tronquées à partir de relations de voisinage extraites des données ; une méthode de dépliage voisine est le *Locally Linear Embedding* [ROW 00]. L'ACP à noyaux (Kernel PCA) [SCH 98] étend implicitement l'espace des descripteurs au moyen d'une fonction noyau choisie par l'utilisateur, et réalise une ACP classique dans cet espace enrichi. Elle capte ainsi les effets des combinaisons de plus de deux variables qui échappent aux approches classiques (effets d'interaction), au bénéfice de la clarté et de la pertinence de la représentation.

La clarté supérieure constatée des représentations issues de ces méthodes se paye d'un inconvénient important : on perd la notion de dualité entre l'analyse des lignes et celle des colonnes, qui permettait d'illustrer, d'expliquer, les dimensions trouvées par les méthodes linéaires. Ceci peut être gênant, ou pas, selon les applications.

Les *réseaux neuronaux non supervisés* : ce formalisme recouvre une vaste famille d'algorithmes qui régissent l'évolution et l'interaction de "cellules" élémentaires dites neurones.

Chaque neurone est caractérisé par un vecteur "poids synaptiques" **m**, à raison d'un poids attribué à chaque "entrée" (= variable) ; la présentation à ce neurone d'un vecteur-individu **x** entraîne une valeur de sortie η' fonction croissante de l'"activité" η du neurone (produit scalaire $\eta = \langle \mathbf{x}, \mathbf{m} \rangle$ du vecteur-individu et du vecteur-poids), et une modification des poids, dite "apprentissage".

Cet apprentissage est généralement de type Hebbien, c'est-à-dire qu'il consiste, pour un neurone isolé, sans contrainte, en une montée en gradient sur une fonction objectif, par exemple ici la somme, pour tous les vecteurs-individus, des carrés des sorties :

$\mathbf{m}(t + 1) = \mathbf{m}(t) + \alpha\, \eta' \mathbf{x}$, où α est une constante petite (par ex. 1/1000), avec une normalisation périodique de **m**.

Les fonctions de transfert « sortie η' en fonction de l'activité η » peuvent prendre diverses formes, entre autres :

- fonction identité : on démontre [OJA 82] que le vecteur-poids converge alors vers le premier vecteur singulier de la matrice des données (fonction objectif : inertie = somme des η^2 = somme des carrés des projections des vecteurs-données sur l'axe **m**).

Cette fonction est présente dans le modèle K-Means Axiales [LEL 91] à fonction objectif pour chaque neurone : inertie locale = somme des η^2 des données propres à la classe correspondante ; ce modèle maximise la somme des inerties locales, pour K donné.

- fonction à seuil, par exemple $\eta' = \eta - \eta_0$ si $\eta > \eta_0$, où η_0 est une valeur de seuil.
  $\eta' = 0$ sinon

Cette fonction est présente dans notre autre modèle Analyse en Composantes Locales [LEL 89] à fonction objectif pour chaque neurone : inertie locale = somme des η'^2 ; se référer aussi à [OJA 91].

- fonction logistique (saturation) - voir [JUT 88] pour la méthode de séparation aveugle de signaux ICA (*Independant Components Analysis*), où la contrainte d'indépendance des composantes est prise dans un sens plus exigeant que celui utilisé dans l'ACP, et dont la variante *Fast ICA* a été utilisée pour extraire avec succès des facteurs-thèmes dans des corpus textuels [TAN 95].

Dans la même lignée, l'analyse en composantes curvilignes (CCA) [DEM 97] est un algorithme neuronal de dépliage dont la fonction objectif suit les principes exposés plus haut pour le MDS (privilégier les voisinages proches au détriment des voisinages éloignés).

Des structures d'inhibition/excitation particulières (en grilles 2D à mailles carrées, triangulaires, ...) caractérisent le modèle de carte auto-organisatrice (SOM : *Self-Organizing Map*) très utilisé et étudié de Kohonen [KOH 95], qui réalise ainsi simultanément l'apprentissage des données et la cartographie d'ensemble positionnant les neurones entre eux (voir figure 5.7).

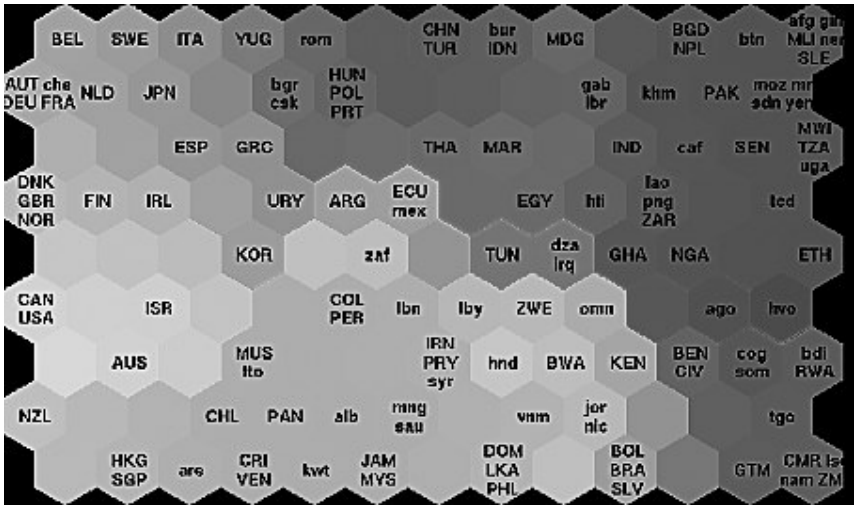

Figure 5.7. Cartographie auto-organisatrice des pays du monde, selon divers indicateurs de pauvreté. Source : Neural Networks Research Centre, Helsinki University of Technology, Finland.

D'autres méthodes donnent des résultats formellement proches de ceux des réseaux neuronaux non supervisés, en abandonnant la contrainte d'orthogonalité des axes factoriels classiques. Ainsi :

- La *poursuite par projection* (" projection-pursuit ") [FRI 94] On recherche ici une ou plusieurs directions "intéressantes" sur lesquelles projeter le nuage de points, après centrage, réduction et "sphérage" des données par ACP (même variance unité dans toutes les directions). Une direction étant d'autant plus intéressante que la

répartition des projections s'éloigne de la loi normale[14], on maximise un indice de non-gaussianité, par exemple la kurtosis κ (excès d'aplatissement) :

$\kappa = E(\eta^4) - 3$, où $E(\eta^4)$ est le moment centré-réduit d'ordre 4, de valeur nulle pour une répartition de Gauss.

- D'autres, comme NNMF (Non-Negative Matrix Factorization, [LEE 99]), ou PLSA (Probabilistic Latent Semantic Analysis [HOF 99]) relâchent les contraintes de centrage-réduction des données. Cette famille de méthodes débouche sur des modèles plus complexes explicitant par exemple un processus de choix de mots dans un ensemble de textes appartenant de façon floue à plusieurs thèmes [BUN 02], où l'estimation des composantes se fait au moyen d'algorithmes de type EM (Expectation Maximization) ; certains réalisent ainsi une décomposition parallèle multiplicative du tableau de données, au lieu de la décomposition séquentielle additive décrite plus haut pour l'ACP.

5.3.3. *La voie du clustering*

Les dernières méthodes citées plus haut, à partir (et y compris) des réseaux neuronaux, ne concernent pas la construction directe d'une représentation des items de base originels (documents et mots) dans un espace 2D ou 3D. Leur intérêt est de donner matière à une représentation à deux étages (carte globale, puis axes ou cartes locaux) adaptée à la taille des données aujourd'hui disponibles, brisant ainsi le mur de la centaine d'items raisonnablement affichables graphiquement sur un écran informatique standard. En ce sens, elles se rapprochent de certaines méthodes de clustering présentées ci-après, dont elles constituent une version floue et recouvrante :

- *Les méthodes à centres mobiles* :

Le prototype de cette famille est la méthode des K-means qui, malgré ses plus de 40 ans d'existence et ses innombrables variantes, reste très populaire car elle réalise un bon compromis entre qualité du résultat, rapidité de calcul et exigence en mémoire vive. Rappelons son principe : on choisit un nombre maximum K de clusters à créer, on sème au hasard K centres de classes (dits aussi prototypes ou

[14] De nombreux travaux ont montré que les répartitions gaussiennes, concentrées autour de la moyenne, sont inadaptées pour rendre compte de beaucoup de phénomènes du vivant (fréquences des mots, taille des villes, interactions entre gènes…) et aussi du monde physique (taille des lacs, longueur des rivières…) [BAR 93]. Alors que l'ACP est conçue pour des mesures à répartition gaussiennes sur une collection d'objets, certaines des autres méthodes citées sont mieux adaptées par nature au type de données à répartition très inégalitaire qui nous intéresse ici, où les valeurs extrêmes sont loin d'être rares.

centroïdes), par exemple en tirant K documents au hasard dans le corpus, puis chaque nouveau document est attribué à la classe N° k dont le centre est à la distance d_k la plus proche, ce qui a pour conséquence de rapprocher ce centre vers le point-document d'une longueur $d_k /(n_k + 1)$ où n_k est le nombre de documents déjà attribués à la classe k.

Cette méthode minimise la fonction objectif : somme des variances intra-classes, où la variance interne à une classe est définie comme la somme des carrés des distances des éléments de cette classe à son centre. Mais le minimum atteint est local, et dépend de la configuration initiale des centres[15], ce qui empêche de garantir l'unicité et la stabilité des résultats. Cette limite est commune par construction à toutes les méthodes à centres mobiles, et peut être gênante dans nombre de contextes d'application.

- *Les méthodes hiérarchiques* :

Ces méthodes consistent à construire progressivement un arbre binaire, ou dendrogramme par regroupement progressif des éléments les plus semblables. L'algorithme de base de la CAH (Classification Ascendante Hiérarchique) est le suivant : on part de la matrice carrée des distances, déduite de celle des similarités décrite plus haut ; on agrège les 2 éléments les plus proches. Si on nomme d_1 leur distance, on dit qu'ils s'agrègent au niveau d_1. Puis on crée une matrice de distance, plus petite d'une ligne et une colonne, en calculant les distances du nouvel ensemble agrégé aux autres éléments (ceci nécessite d'avoir choisi une méthode pour cela, par exemple la distance du centre de gravité de la nouvelle classe aux autres éléments), et on continue le processus jusqu'au regroupement final en un seul ensemble. L'avantage est que le dendrogramme peut se construire en parallèle, en même temps que le processus d'agrégation, sans problème de croisement de lignes puisque le niveau d'agrégation ne peut que croître par définition, et que tout élément nouveau se place soit à la suite, soit par simple insertion dans la liste des éléments déjà classés (voir figure 5.8).

[15] Il dépend aussi de l'ordre de présentation des documents pour la version présentée ci-dessus ; mais cette contrainte est facilement levée par les variantes itératives des K-means.

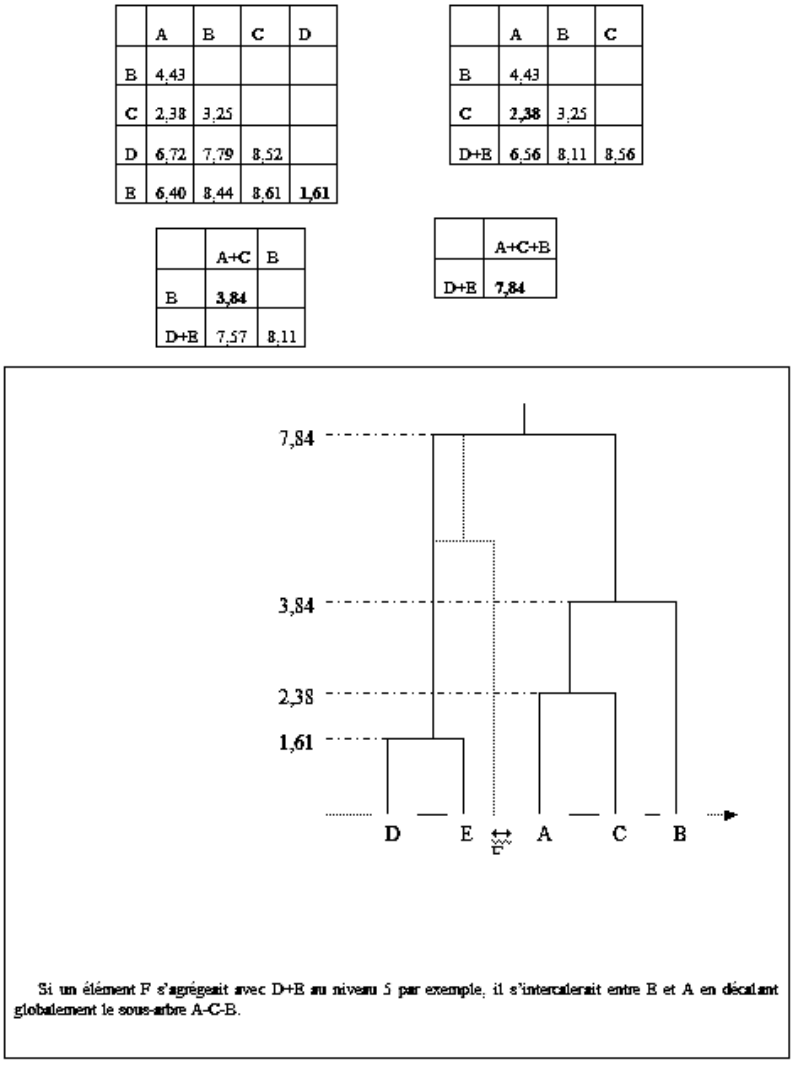

Figure 5.8. *Principe de la classification ascendante hiérarchique : construction et visualisation*

Un autre avantage est que l'on visualise bien les différentes façons de découper les données en clusters, par la définition d'un seuil de coupure, en terme de niveau d'agrégation, dans les zones les moins « touffues » de l'arbre.

- *Les méthodes à représentations arborées* :

Ce type de représentation, à la fois esthétique et rigoureux, repose sur le même principe que l'échelonnement multidimensionnel, à savoir : reproduire dans le plan un ensemble de distances (ou plus généralement : de dissimilarités) entre points d'un nuage multidimensionnel de la façon la plus fidèle possible [BAR 91]. Mais la distance dans le plan visée ici n'est pas la distance euclidienne, ordinaire, du plus court chemin entre deux points, c'est une distance mesurée par le chemin à parcourir le long des arêtes d'un arbre entre deux de ses feuilles (voir figure 5.9).

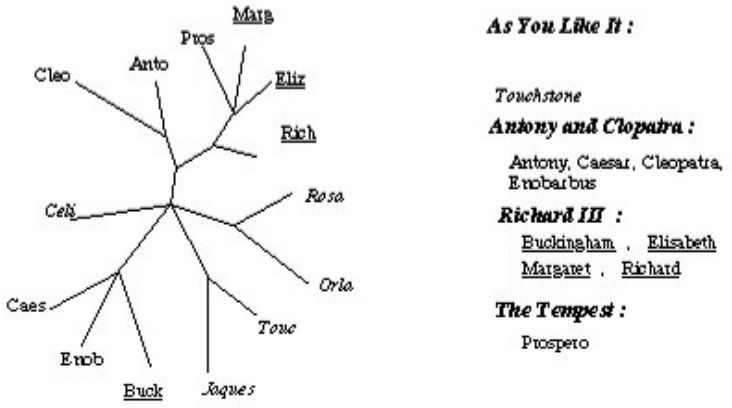

Figure 5.9. Représentation arborée des similarités de langage (à partir des 40 mots les plus fréquents) entre 14 personnages de Shakespeare. Source : JP. Barthélémy et JADT'1998

Malheureusement la lourdeur des calculs nécessaires (au moins en $O(N^4)$) limite son application à la représentation d'ensembles de quelques dizaines d'éléments au plus. Un des principes utilisés est de créer par CAH, comme vu ci-dessus, un arbre-support sur lequel on minimise la somme des carrés des différences entre distances (ou dissimilarités) vraies et distances arborées.

5.3.4. *La voie des graphes*

Une matrice de similarité (individus × individus) peut être construite à partir de toute définition de la similarité, et traduite sous forme de graphe dont les noeuds sont les individus et les arêtes des fonctions des valeurs rencontrées, par exemple une fonction de seuil : au-dessus du seuil choisi, il existe une arête, au dessous, non. La matrice de similarité devient une matrice d'adjacence de graphe ne comportant des uns (ou des valeurs positives non nulles) que lorsque des liens existent. Cette notion de graphe incarne un degré de stylisation supplémentaire dans la représentation de réalités complexes : seule compte l'existence de liens entre individus ; on ne cherchera pas à représenter au mieux l'ensemble de leurs degrés de similarités, mais la topologie de leurs relations, de la façon la plus claire possible.

Quand on dispose directement de ces relations, comme c'est le cas pour les liens entre pages Web, pour les citations émises par un article scientifique ou pour les synonymes d'un mot, il va de soi qu'on peut passer directement au dessin du graphe. Sauf exceptions, ces liens ne sont pas réciproques et on construira alors un graphe orienté, dont la matrice d'adjacence ne sera pas symétrique.

De nombreuses méthodes permettent de projeter au mieux un graphe sur 2 dimensions, ou de le partitionner [BRA 03]. Nous présentons les familles les plus importantes, à savoir les méthodes spectrales, les méthodes de placement à base de forces, et les méthodes d'optimisation d'une fonction objectif orientée « ergonomie visuelle ».

Le degré ultime de stylisation est franchi quand on impose au graphe une topologie en forme d'arbre[16], c'est-à-dire une organisation hiérarchique et emboîtée de la connaissance, avec les avantages et inconvénients que cela comporte : clarté de la vue d'ensemble, mais aussi difficulté à retrouver un élément précis quand on n'est pas familier avec l'organisation choisie. Nous verrons d'abord comment on peut passer d'un graphe quelconque à un arbre, puis les principales méthodes de visualisation d'arborescences.

- *Méthodes spectrales* :

  L'adjectif spectral fait référence à la décomposition en éléments propres (valeurs propres et vecteurs propres) d'une matrice carrée. Ainsi, l'analyse factorielle des correspondances (AFC) de la matrice d'adjacence **A** d'un graphe permet de représenter celui-ci dans le plan des deux (ou volume des trois) premiers facteurs

[16] Dans les arbres étudiés à la section précédente, on attribuait une signification précise à la longueur des branches, sous-branches, etc. ; ici seule compte la configuration topologique de l'arborescence.

non triviaux [LEB 84] : l'AFC de la matrice d'adjacence des départements français reconstitue dans ses grandes lignes la carte de la France... Cette méthode a été retrouvée plus récemment [BRA 03], et formulée comme l'extraction des deux plus petits éléments propres non nuls du laplacien **L** du graphe :

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

où **D** est la matrice diagonale des degrés du graphe.

Ces méthodes n'optimisant pas spécifiquement le nombre de croisements d'arêtes, elles sont bien adaptées à la représentation tridimensionnelle des graphes, où le problème ne se pose pas. Elles ont aussi l'avantage de la rapidité, de la stabilité et de la reproductibilité : à une matrice d'adjacence ne correspond qu'une seule représentation, qualité appréciable dans certains types d'applications, où un « fond de carte » stable est nécessaire.

- *Méthodes à relaxation de contraintes* (« masses et ressorts »).

Le placement orienté par les forces (FDP, *Force Directed Placement*) [EAD 84] consiste à créer un système mécanique virtuel, image du graphe à représenter, en considérant les nœuds comme des masses qui se repoussent, et les arêtes comme des ressorts qui attirent les couples de nœuds liés (voir figure 5.10).

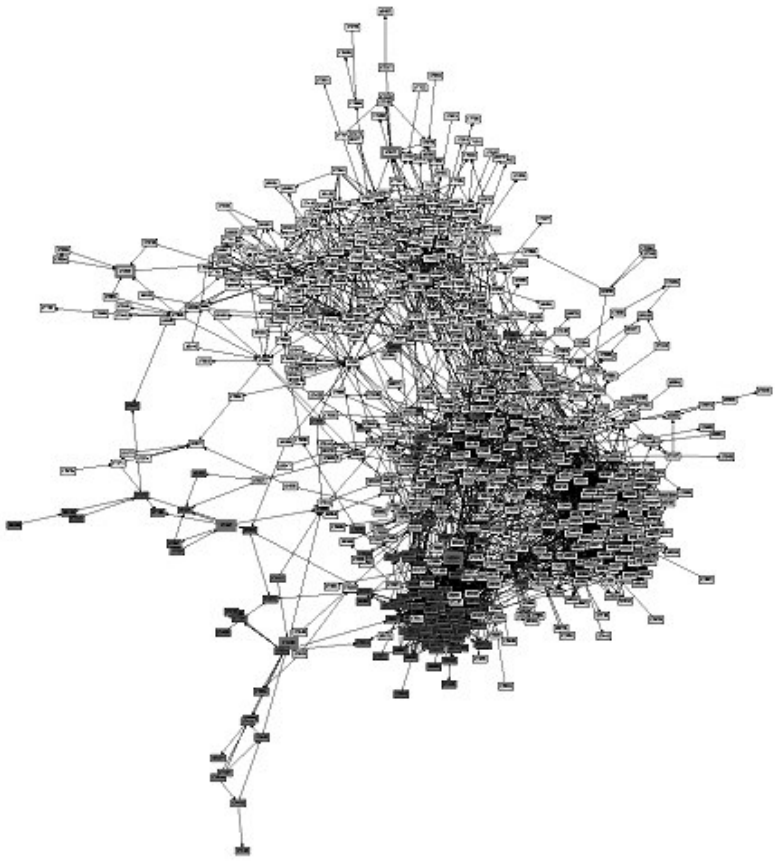

Figure 5.10. *Représentation « masses et ressorts » de résumés d'articles géotechniques de la base Pascal à partir du logiciel libre AiSee. Source : INIST, Nancy*

Voici le principe de l'algorithme *Spring Embedder* : on place au hasard dans le plan les nœuds du système-image. Pour chaque nœud on calcule la résultante des forces de rappel (de la part des nœuds liés) et de répulsion (de la part de tous les nœuds) et on lui imprime un léger déplacement dans cette direction proportionnellement à l'intensité. Puis on itère jusqu'à stabilisation. Des applets Java comme celle qu'utilise TouchGraph permettent de visualiser en temps réel et de façon spectaculaire ce processus « physique » et rapide de relaxation de

contraintes, dont l'inconvénient est la dépendance par rapport à l'initialisation, donc la non-reproductibilité pratique.

- *Optimisation d'une fonction indicatrice de qualité d'ergonomie visuelle*.

On peut définir un ou plusieurs indicateurs de qualité de la représentation, comme le nombre de croisements d'arêtes (à minimiser), ainsi que des contraintes à respecter, comme la distance minimale entre deux nœuds. La fonction objectif globale à optimiser (« énergie ») sera définie comme une somme, pondérée à volonté, des indicateurs partiels. Des techniques génériques d'optimisation, comme le recuit simulé ou les algorithmes génétiques peuvent alors être mises en œuvre : en partant d'une disposition des nœuds au hasard, le recuit simulé conserve les déplacements qui font baisser l'énergie tout en respectant les contraintes, et autorise les autres déplacements avec une certaine probabilité (dite « température »), paramètre qu'on fait baisser progressivement jusqu'à stabilisation de la fonction objectif. Quant aux algorithmes génétiques, ils créent et font évoluer une *population* de solutions, et non une seule, par mutations aléatoires et croisements des meilleures solutions entre elles.

Comme cette description le laissait prévoir, les représentations obtenues peuvent être de qualité excellente, mais au prix d'un temps de calcul élevé et d'une mise au point des paramètres qui est tout sauf presse-bouton.

On peut noter aussi que l'optimalité de la solution n'est pas toujours souhaitable : dans des applications dynamiques ou interactives où l'on ajoute à chaque mise à jour un nombre limité de nœuds et de liens, il faut préserver la « carte mentale » de l'usager en n'apportant que de légères corrections à l'état précédent de la représentation, et privilégier la continuité plutôt que la qualité.

- *Simplifier les graphes : les arbres.*

Des graphes aux arbres :

Tout graphe, avec ou sans valuation des arêtes, peut être stylisé sous forme d'arbre. L'avantage est qu'un arbre admet par essence une représentation planaire, et qu'il y a peu de contraintes de placement pour ses éléments, car sa topologie reste visuellement évidente pour une large variation de la longueur de ses embranchements. Un résultat mathématique important est que toute composante connexe d'un graphe peut être élaguée sous forme d'arbre dit à recouvrement minimum (*Minimum Spanning Tree*) dans lequel l'ensemble de ses nœuds sera présent. Voici un algorithme simple pour y parvenir :

. partir d'un nœud quelconque du graphe, qu'on marque et qui forme l'élément initial A_0 de l'arbre.

. à chaque itération on construit l'arbre A_t en augmentant A_{t-1} de l'arête (ou des arêtes) de similarité maximum dont une seule extrémité est marquée. On marque alors l'extrémité non marquée de l'arête (ou des arêtes) sélectionnée(s) et on itère le processus jusqu'au marquage de l'ensemble des nœuds.

A noter qu'en cas de similarités *ex-aequos* (par exemple si tous les liens sont valués 1), la solution n'est pas unique et les embranchements peuvent ne pas être binaires.

Beaucoup de travaux ont été consacrés à la visualisation interactive des très grandes structures arborescentes. Nous en présentons ici deux classes caractéristiques, les vues hyperboliques et la représentation sous forme de rectangles emboîtés.

Visualisation hyperbolique

Pour naviguer dans un grand graphe planaire, il peut être intéressant de focaliser sur un des nœuds du graphe sans perdre de vue le contexte d'ensemble : une vue « fish-eye » peut être créée en appliquant une transformation hyperbolique à l'ensemble des nœuds et arêtes du graphe, centrée sur le nœud sélectionné. Par exemple, on est clair que sur la demi-droite positive, la transformation simple $x \leftarrow x/(1 + x)$ transforme peu les points voisins du nœud-foyer origine, transforme 1 en ½, 2 en 2/3, etc. et $+\infty$ en 1.

La façon la plus directe de rendre planaire un grand graphe étant de le styliser sous forme d'arbre, ce procédé rapide de « transformation optique virtuelle » a été appliqué à la visualisation interactive de grandes arborescences issues du Web (forums, messageries, liens entre pages…) – voir figure 5.11.

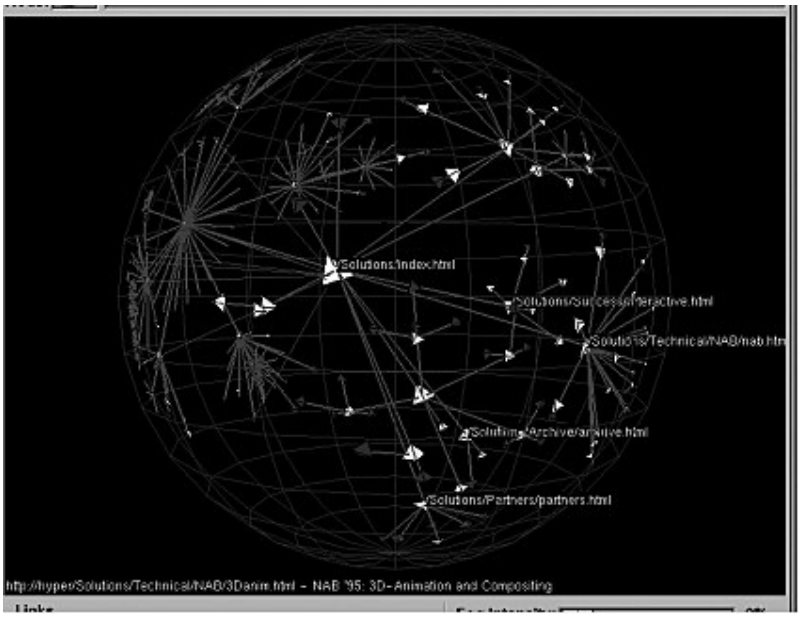

Figure 5.11. *Vue hyperbolique de l'arborescence d'un site Web, par Site Manager. Source : Silicon Graphics et cybergeography.org*

Tesselation d'arbres (*TreeMap* [BEN 92]) :

Ce mode de représentation original permet de concrétiser visuellement les relations d'inclusion incarnées par une arborescence, tout en autorisant une focalisation interactive sur une branche, quel que soit son niveau. En effet tout arbre peut être représenté par un emboîtement de rectangles correspondant aux différents niveaux de sa hiérarchie. Principe général : le rectangle d'origine est subdivisé, verticalement et horizontalement en alternance, proportionnellement au « poids » en feuilles de chaque sous-arbre représenté – voir figure 5.12.

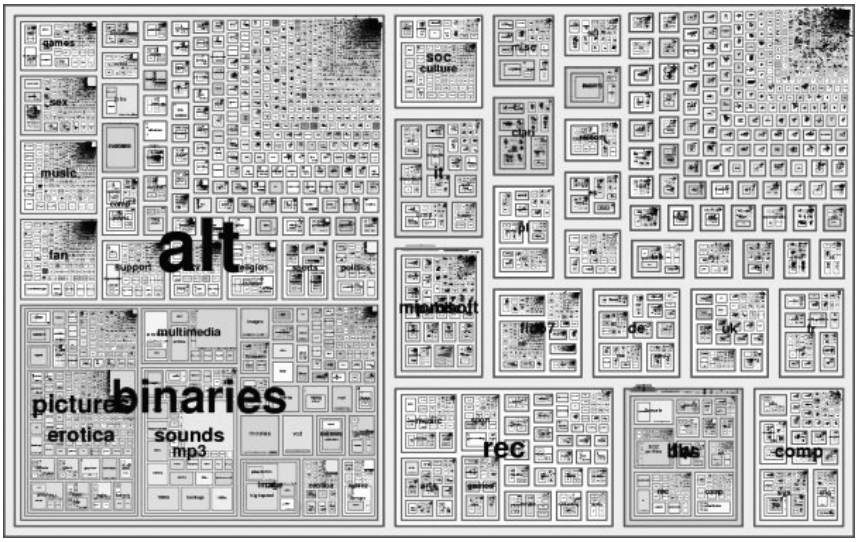

Figure 5.12. *Représentation d'une arborescence Usenet par tesselation [Bernheim et al 05]*

5.3.5 Choix d'une méthode

Comme on vient de le voir, chaque méthode de représentation a ses avantages et ses inconvénients. Voici, sans prétention à l'exhaustivité, quelques points saillants en matière de choix. Nous ne traiterons pas les points à caractère informatique fort, au-delà des considérations de temps de calcul, qui sont bien sûr importants et parfois déterminants en pratique, comme la disponibilité dans le commerce ou gratuitement de modules logiciels d'analyse et visualisation, leur possibilité d'intégration dans la réalisation envisagée, leur caractère fermé ou ouvert, les compétences qu'ils demandent pour la réalisation du projet, sa maintenance, son évolution…

- Pour la visualisation de moins d'une centaine de documents textuels, les méthodes factorielles classiques (ACP, AFC…) sont adéquates et possèdent sur les autres méthodes la supériorité de pouvoir représenter sur la même carte la position des mots, permettant d'expliquer à l'utilisateur le pourquoi des regroupements constatés. Même si le placement des points n'est pas visuellement optimal (mais améliorable par l'heuristique des rangs factoriels), elles présentent un caractère « indiscutable » - à un tableau de données correspond à peu de choses près une seule carte, quel que soit l'ordinateur, le logiciel de traitement ou l'endroit du monde… De ce fait elles constituent d'excellents « fond de carte » pour faire apparaître dynamiquement des liens entre les textes ou les mots à divers seuils de similarité, ou pour placer de façon passive, par simple projection, de nouveaux points-documents

supplémentaires dans une carte existante. Ces cartes peuvent être « éclairées » par des variables catégorielles internes ou externes à l'analyse, représentables également par les centres de gravité des documents qui appartiennent à telle ou telle catégorie.

Par comparaison, les méthodes de projection non-linéaires directes, comme le MDS, l'ADI, le LLE ou le Kernel PCA fournissent des cartes *a priori* plus claires, mais perdent l'avantage de l'explication simultanée par les descripteurs.

Si les méthodes précédentes mettaient l'accent sur la visualisation la plus fidèle possible en 2D ou 3D de similarités dans un espace à un grand nombre de dimensions, les méthodes de graphe élaguent radicalement le problème en ne considérant que des liens, en tout ou rien, et en portant tout l'effort sur la clarté visuelle de la représentation. Elles délèguent souvent à l'utilisateur la possibilité de l'améliorer encore en modifiant à la souris la place des nœuds du graphe et des libellés. Des systèmes de couleurs, icônes, pop-ups apparaissant au survol de la souris permettent de les enrichir en informations internes (mots) ou externes à l'analyse (auteur, année,…). De gros efforts de design ont été faits dans ce sens par des moteurs de visualisation comme Kartoo[17], TouchGraph ou Mapstan. Le problème est alors de se prémunir contre la surcharge cognitive provoquée par une trop grande richesse de symboles, dont le sens ne peut être raisonnablement présent à l'esprit de l'utilisateur qu'au bout d'un véritable apprentissage.

- Pour des nombres de documents importants (disons de 100 à un million) les techniques de visualisation de très grands graphes sont disponibles. Elles sont adéquates quand il s'agit de localiser un élément précis, puis zoomer et dézoomer à volonté sur son contexte – cas envisageable pour les pages et sites Web, par exemple. Mais l'autre voie est celle de la création d'un niveau intermédiaire de représentation qui fasse sens pour l'utilisateur, qui lui présente les grandes tendances à l'œuvre dans les données, appréhendables d'un seul coup d'œil sur la carte globale : étant juge et partie en cette affaire, nous nous contenterons de signaler qu'outre notre méthode des K-means axiales [LEL 91], on trouve dans cette catégorie les cartes auto-organisatrices de Kohonen, avec de nombreuses réalisations éprouvées, et potentiellement toute méthode rattachable formellement aux représentations factorielles obliques ou au clustering flou, comme les NNMF, ICA, CCA et autres PLSA.

5.4. Conclusions

La cartographie de l'information s'inscrit dans une progression de fonctions d'interaction offerte à l'utilisateur, des plus élémentaires aux plus complexes :

[17] <www.kartoo.com>

1) en premier lieu : choisir dans une courte liste de possibilités – on peut dire que la simple présentation d'un menu d'items à sélectionner a fait l'objet d'autant de variantes esthétiques, des plus spartiates aux plus baroques, que le thème de la chaise dans le design industriel… 2) Puis choisir un élément parmi un très grand nombre : c'est là que la puissance du langage peut intervenir, sous forme de zone de saisie de mots. Mais ce type de requête ne peut être que ponctuelle, alors que 3) l'approche graphique permet, elle, de proposer en alternative une vue d'ensemble, que l'utilisateur peut explorer et approfondir à loisir. Jointe aux possibilités croissantes d'interactivité, elle permet de représenter plusieurs ensembles en relation (ex., pour des articles : mots, auteurs, pays, …) sous forme de coloriages et liens ; de plus en plus nombreux environnements de visualisation et métamoteurs Web comme TouchGraph, Kartoo ou Pajek (voir figure 5.13), font usage de ces possibilités.

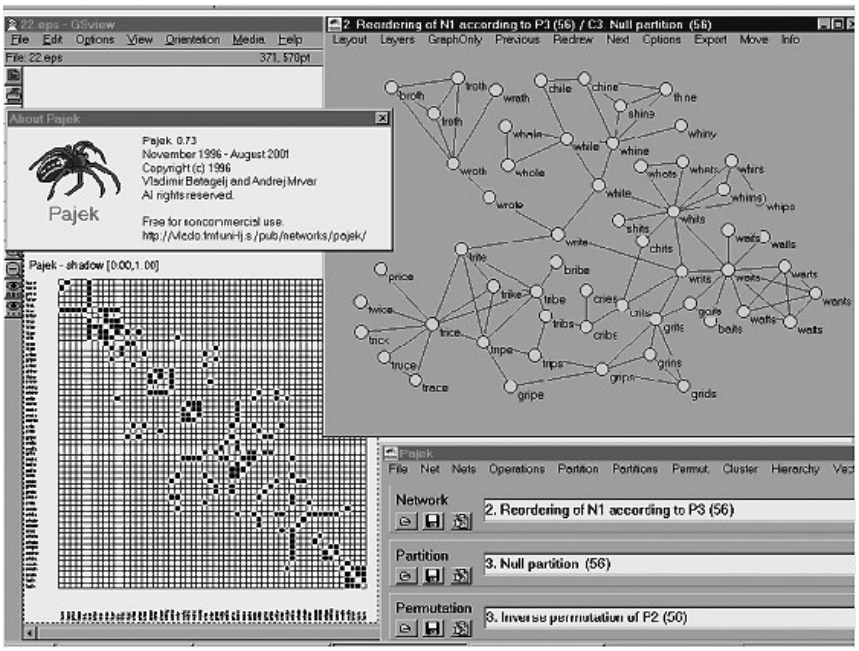

Figure 5.13. *Visualisation PAJEK de relations entre formes lexicales*

Une voie de recherche porte désormais sur l'introduction du temps dans ces représentations, car les réalités auxquelles nous sommes confrontés sont fondamentalement dynamiques ; il s'agit, pour faire court, de passer de l'image fixe au dessin animé [BAT 05] (voir figure 5.14)… Ce défi n'est pas sans lien avec les tendances lourdes de l'informatique : alors que les architectures centralisées des

réseaux d'autrefois ont cédé définitivement la place aux architectures décentralisées et robustes comme Internet, l'étape suivante consiste à transposer le traitement des immenses réservoirs de connaissances vers un mode réparti et dynamique. Les représentations visées, visuelles en particulier, devront s'adapter en permanence, par un dialogue au sein d'un réseau de serveurs et « serveurs de services », aux modifications et ajouts de données nouvelles. Notre algorithme GERMEN [LEL 06] constitue une première tentative dans le sens d'une cartographie décentralisée et adaptative de ressources multiples réparties.

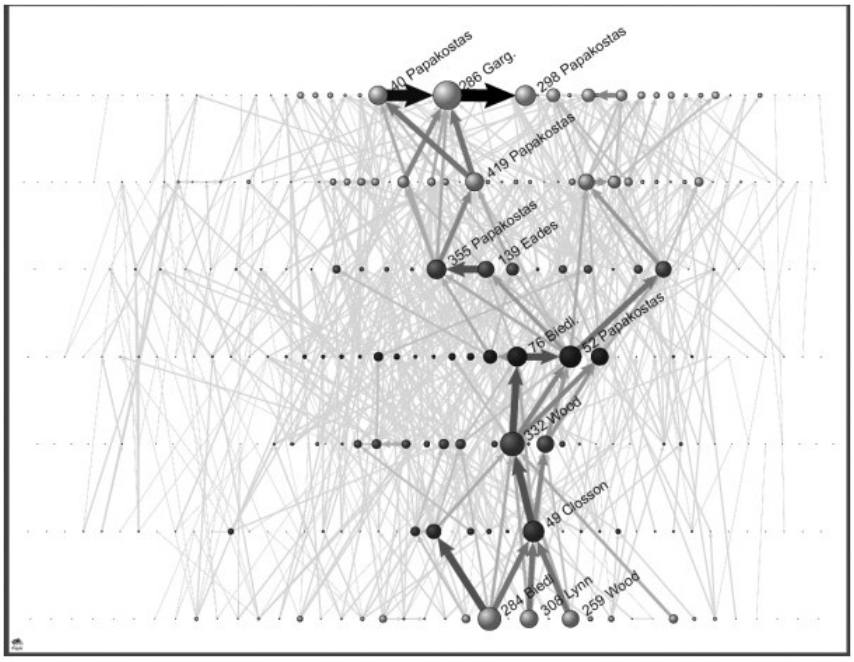

Figure 5.14. *GD 2005 Conference – Evolving Graph Drawing Contest 2005. Layouts by Vladimir Batagelj and Andrej Mrvar, University of Ljubljana, Slovenia*

De même que le design industriel combine au mieux les possibilités et les contraintes des matériaux, des procédés de fabrication (artisanaux ou à grande échelle), des aspects économiques (modularité, maintenance, …), à des possibilités et contraintes d'usage des objets, aboutissant à un équilibre plus ou moins réussi – seul le succès d'usage peut le dire - , de même le design des interfaces d'accès graphique à une information encore majoritairement textuelle, organise 1) le choix des méthodes de réduction de dimension en fonction de la taille des données, de la

taille du vocabulaire retenu, des contraintes d'efficacité informatique (mise à jour en temps réel ou pas, interactivité, puissance des processeurs) et des objectifs attribués aux utilisateurs[18], 2) le choix des fonctions mises à disposition de ceux-ci, 3) l'ergonomie du dialogue : quand et où apparaissent menus, pop-ups et ascenseurs, quelles possibilités de zoom, retour en arrière, ou de déposer de l'information en retour… .La palette des possibilités offertes au designer par l'informatique et par les méthodes de réduction de dimensions ne cesse de s'enrichir. Les illustrations de cet article témoignent que les concepteurs et designers savent en profiter pour nous proposer des visualisations d'information à la fois fonctionnelles et puissamment esthétiques (voir figure 5.15).

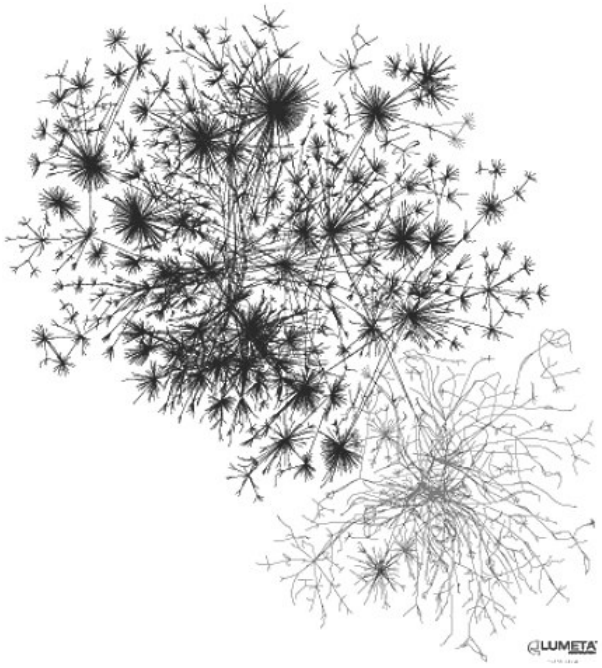

Figure 5.15. *Internet Mapping Project - Bill Cheswick. Source : LUMETA et cybergeography.org*

[18] que ceux-ci s'empressent de déborder, l'usager est un pirate, c'est bien connu… L'équilibre entre usage prévu et usage réel n'est jamais donné à l'avance, comme le rappellent aux enseignants les multiples et improbables utilisations de Google et Wikipedia par leurs étudiants…

5.5. Bibliographie

[BAR 03] BARBUT M., « Homme moyen ou homme extrême ? De Vilfredo Pareto (1895) à Paul Lévy (1935) en passant par Maurice Fréchet et quelques autres », *Journal de la Société Française de Statistique*, vol. 144, n° 1-2, 2003.

[BAR 91] BARTHÉMEMY J.P., GUÉNOCHE A. *Trees and Proximity Representation,* New-York, Wiley & son, 1991.

[BAT 01] BATAGELJ V., MRVAR A., « Pajek-Analysis and Visualization of Large Networks », *Graph Drawing: 9th International Symposium*, Vienna, 2001.

[BAT 05] BATAGELJ V., MRVAR A., in Evolving Graph Contest, *Graph Drawing: 13th International Symposium*, 2005.

[BEN 80] BENZECRI J.P. *Pratique de l'analyse des données*, Tome 1, Dunod, 1980.

[BEN 81] BENZECRI J.P. ET COLL. *Pratique de l'Analyse des Données*, Tome 3 : Linguistique et Lexicologie, Dunod, Paris, 1981.

[BER 05] BERNHEIM BRUSH A.J., WANG X., COMBS TURNER T., SMITH M. A., « Assessing Differential Usage of Usenet Social Accounting Meta-Data », CHI 2005, p. 889-898, 2005.

[BRA 03a] BRANDES U., GAERTLER M., WAGNER D. « Experiments on Graph Clustering Algorithms », *Proceedings of the 11th Europ. Symp. Algorithms (ESA '03)*, Springer LNCS, 2003.

[BRA 03b] BRANDES U., COMELSEN S., « Visual Ranking of Link Structures », *J. Graph Algorithms and Applications*, 7(3) p. 179-199, 2003.

[BUN 02] BUNTINE W.L., « Variational extensions to EM and multinomial PCA », *13th European Conference on Machine Learning (ECML'02)*, Helsinki, Finland, 2002.

[BUR 91] BURTSCHY B., LEBART L., « Contiguity analysis and projection pursuit », *Appl. Stoch. Mod. and Data Anal.,* Gutierrez R. et al. Eds, World Scientific, p. 117-128, Singapore, 1991.

[CAD 07] CADOT M., CUXAC P., LELU A., « Random simulations of a datatable for efficiently mining reliable and non-redundant itemsets », *Applied Stochastic Models and Data Analysis*, Chania, Grèce, 2007.

[DEM 97] DEMARTINES P., HÉRAULT J., Curvilinear Component Analysis: a Self-Organising Neural Network for Non-Linear Mapping », *IEEE Trans. on Neural Networks*, 8(1):148-154, 1997.

[EAD 84] EADES P. « A heuristic for Graph Drawing », *Congressus Numerantium,* vol. 42, p. 149-160, 1984.

[FRI 74] FRIEDMAN J.H., TUKEY J.W., « A projection pursuit algorithm for exploratory data analysis », *IEEE Transactions on Computers,* C-23, 881, 1974.

[HER 04] HERLOCKER J.L., KONSTAN J.A., TERVEEN L.G., RIEDL J.T., « Evaluating collaborative filtering recommender systems », *ACM Trans. Inf. Syst.*, 22 5—53, 2004.

[HOF 99] HOFMANN T., « Probabilistic Latent Semantic Indexing », *SIGIR 1999*: 50-57, 1999.

[HOT 33] HOTELLING H., « Analysis of a complex of statistical variables into principal components », *The Journal of educational psychology*, vol. 24, p. 417-441 et 498-520, 1933.

[JUT 88] JUTTEN C., HERAULT J., « Une solution neuromimétique au problème de séparation de sources », *Traitement du Signal,* Vol. 5, N° 6-NS, p. 389-403, 1988

[KOH 95] KOHONEN T., *Self-Organizing Maps*, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 501 pages, 1995.

[KRU 64] KRUSKAL J.B., « Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis », *Psychometrika*, 29, 1-28, 1964.

[LEB 84] LEBART L., « Correspondence analysis of graph structures », *Bulletin technique du CESIA*, vol.2, n°1-2, pp. 5-19, 1984.

[LEE 99] LEE D. D. AND SEUNG H. S., « Learning the parts of objects by nonnegative matrix factorization », *Nature*, 401(1999), 788-791.

[LEL 04], LELU A., « Analyse en composantes locales et graphes de similarité entre textes », *Actes de JADT 2004*, G. Purnelle ed., Université catholique de Louvain, 2004.

[LEL 06] A. LELU, CUXAC P., CADOT M., « Document stream clustering : an optimal and fine-grained incremental approach », *COLLNET'06 / International Workshop on Webometrics, Informetrics and Scientometrics*, Nancy, 10-12 mai 2006.

[LEL 97] LELU A., TISSEAU-PIROT A.G., ADNANI A., « Cartographie de corpus textuels évolutifs : un outil pour l'analyse et la navigation », *Hypertextes et Hypermédias,* vol.1, N°1, éditions Hermès, Paris, 1997.

[LEL 89] LELU A., « Local Component Analysis : a neural model for information retrieval », *Actes de l'International Joint Conference on Neural Networks* - Washington, vol. II p.43-48, IEEE, 1989.

[LEL 91] LELU A., « From data analysis to neural networks : new prospects for efficient browsing through databases », *Journal of Information Science*, vol. 17, pp.1-12, Elsevier ed., Londres, 1991

[LEL 94] LELU A., « Clusters and factors: neural algorithms for a novel representation of huge and highly multidimensional data sets », *New Approaches in Classification and Data Analysis*, Diday E., Lechevallier Y. et al. eds., pp.241-248, Springer-Verlag, Berlin, 1994.

[LEL 00a] LELU A., HALLAB M., « Consultation floue de grandes listes de formes lexicales simples et composes : un outil preparatoire pour l'analyse de grands corpus textuels », *Actes de JADT 2000*, coord. : M. Rajman, EPFL, Lausanne, 2000.

[LEL 00b] LELU A., HALLAB M., PAPY F., BOUYAHI S., RHISSASSI H., BOUHAÏ N., TANG F., « Textual mapping for multilingual and multiwriting access to information on the Internet », *Actes de RIAO 2000*, coord. CID, Collège de France, Paris,12-14 Avril 2000.

[NAD 06] NADEAU D., TURNEY P., MATWIN S., « Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity », *19th Canadian Conference on Artificial Intelligence*, Québec City, Québec, Canada, 2006.

[OJA 91] OJA E., OGAWA H., AND WANGVIWATTANA J., « Learning in nonlinear constrained Hebbian networks », in T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (Eds.), *Artificial Neural Networks,* Elsevier, pp. 385 - 390, Amsterdam, 1991

[PEA 01] PEARSON K., « On lines and planes of closest fit to systems of points in space », *Phil. Mag*, n° 2 (6ème série), p. 559-572, 1901.

[RAJ 00] RAJMAN M., BESANÇON R., CHAPPELIER J.-C., « Le modèle DSIR : une approche à base de sémantique distributionnelle pour la recherche documentaire », *Traitement automatique des langues*, vol. 41, N° 2, p. 549-578, 2000.

[ROW 00] ROWEIS S.T., SAUL L.K., « Nonlinear dimensionality reduction by locally linear embedding ». *Science*, 290(5500):2323–6, 2000.

[SCH 98] SCHÖLKOPF B., SMOLA A., MÜLLER K.-R., « Nonlinear component analysis as a kernel eigenvalue problem », *Neural Computation*, 10:1299–1319, 1998.

[SPE 04] SPEARMAN, C. E., « 'General intelligence' objectively determined and measured ». *American Journal of Psychology*, 5, 201-293, 1904.

[SHE 72] SHEPARD R.N. ET ALII (eds), *Multidimensional Scaling : Theory and applications in the behavioral Sciences*., Vol. 1: Theory, Seminar Press, New York, 1972.

[SHN 92] SHNEIDERMAN B., « Tree visualization with tree-maps: 2-d space-filling approach », *ACM Trans. Graph*. 11, 1, 92-99, 1992.

[TAN 05] TANG B., SHEPHERD M., MILIOS E., HEYWOOD M., « Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering », *International Workshop on Feature Selection for Data Mining*, Interfacing Machine Learning and Statistics, in conjunction with 2005 SIAM International Conference on Data Mining, Newport Beach, California, 2005.