

LE MODELE VECTORIEL POUR LE TRAITEMENT DE DOCUMENTS

D. Memmi

UQAM

e-mail: memmi dot daniel arobase uqam dot ca

The Vector Space Model for Document Processing

Abstract: *we describe the main notions underlying the vector space model for natural language processing and information retrieval. Fundamental concepts of vector space theory will be defined and basic clustering methods will be explained. We show how to apply the vector space model to the most common document processing tasks. We then discuss the problems of the approach, which we finally try to evaluate.*

Résumé : *nous allons exposer les notions principales du modèle vectoriel pour le traitement du langage naturel et la recherche d'information. Nous décrirons notamment les concepts de base sur les espaces vectoriels et la classification des données, ainsi que les grandes applications du modèle vectoriel au traitement de documents. On discutera aussi des problèmes posés et de la validité de l'approche.*

Introduction

Depuis le début des travaux en Traitement Automatique du Langage Naturel (TALN) on a poursuivi des directions de recherche diverses. On peut notamment distinguer des approches numériques s'appuyant sur probabilités et statistique et des approches syntaxiques liées à la théorie des langages formels. On remarque aussi que l'éventail de recherche va de l'analyse détaillée de phrases isolées à des approches plus globales d'un texte dans son ensemble.

L'approche dominante en TALN a suivi la tradition linguistique en prenant la phrase comme unité fondamentale d'analyse et de traitement. L'analyse syntaxique de la phrase (en utilisant grammaires formelles et automates) a été le plus souvent considérée comme un préliminaire indispensable à l'interprétation sémantique (voir par exemple Winograd 83 ; Sabah 90 ; Abeillé & Blache 97). Les efforts ultérieurs pour traiter des textes dans leur ensemble se sont heurtés à la somme d'efforts nécessaires dans cette approche pour l'analyse des phrases puis leur intégration en un ensemble cohérent.

Dans le même temps se développait une direction de travail relativement indépendante du TALN syntaxique, mais davantage liée aux statistiques et à la recherche documentaire. Elle partait plutôt des nécessités de la classification et recherche de documents (Salton & McGill 83) (Salton & Buckley 94) (Leloup 97), mais aussi de motivations plus générales (Lebart & Salem 94) (Yang 98). D'autre part le renouveau actuel des méthodes de traitement de corpus (T.A.L. 95) (Habert et al. 97) favorise les méthodes numériques.

Cette direction numérique est plus proche des mathématiques, et en particulier des probabilités. Plutôt que de construire des structures syntaxiques, on cherche à calculer les probabilités de cooccurrences entre mots ou expressions. Mais on utilise aussi souvent le "modèle vectoriel". C'est ce modèle que nous allons présenter ici, tout en essayant ensuite de le replacer dans le cadre plus large du TALN et de la linguistique.

On peut appliquer des modèles numériques à l'analyse de phrases individuelles (Charniak 93) (Manning & Schütze 99). Ainsi les grammaires probabilistes et les modèles de Markov reprennent les notions de grammaires formelles et d'automates, en y rajoutant des probabilités de transition associées aux règles ou aux graphes des automates. Ces modèles sont tout à fait

efficaces (notamment en reconnaissance de parole), mais nous ne les détaillerons pas ici. Nous parlerons uniquement de modèles numériques s'appliquant à l'ensemble d'un texte choisi.

Dans l'approche vectorielle en effet, on traite non pas des phrases, mais des textes ou des documents dans leur ensemble, en passant par une représentation numérique très différente d'une analyse structurale, mais permettant des traitements globaux rapides et efficaces. L'idée de base consiste à représenter un texte par un vecteur dans un espace approprié, puis à lui appliquer toute une gamme de traitements vectoriels.

Pour donner un exemple, une application typique consiste à représenter des documents par des vecteurs calculés à partir des mots les plus significatifs présents dans chaque document. Ces vecteurs sont ensuite regroupés par similarité de manière à classer ensemble les documents traitant des thèmes similaires. Cette classification peut alors servir à l'indexation et à la recherche des documents, mais aussi à l'extraction d'informations plus élaborées.

Les notions de vecteur et d'espace vectoriel sont donc fondamentales dans ces méthodes, et nous allons d'abord les préciser. Puis nous passerons aux processus de traitement, et en particulier aux techniques de classification, avant de décrire les grands types d'application. Enfin nous tenterons de discuter et d'évaluer la pertinence de cette approche.

1. Les espaces vectoriels

La théorie mathématique sous-jacente à cette approche est la théorie des espaces vectoriels et plus généralement l'algèbre linéaire (Bourbaki 47) (Halmos 74) (Strang 76) (Johnson et al. 98). C'est un domaine très abstrait mais d'un formalisme relativement abordable, et il est fort intéressant d'en suivre le développement lors des deux derniers siècles (Dorier 95).

La théorie s'est constituée par unification et abstraction progressives de concepts venus à la fois de l'algèbre et de la géométrie, et elle a pu s'appliquer à des domaines très divers. On peut citer notamment l'analyse factorielle des données (Bouroche & Saporta 80) (Jolliffe 86). Mais nous nous contenterons ici de rappeler l'essentiel nécessaire à la compréhension de l'exposé, en évitant des développements trop techniques (pour une introduction, voir Jordan 86).

Il est possible de présenter les espaces vectoriels comme une généralisation de l'espace géométrique ordinaire à trois dimensions. Un espace vectoriel peut avoir un nombre quelconque de dimensions, mais ce n'est qu'au milieu du 19^{ème} siècle que les mathématiciens commencèrent à accepter l'idée d'espaces à plus de trois dimensions. Le nombre de dimensions (*la dimension*) d'un espace vectoriel est alors le nombre minimal d'axes de coordonnées nécessaires pour définir tout point de cet espace. De tels axes sont indépendants entre eux, et la notion d'*indépendance linéaire*, fondamentale en algèbre linéaire, est également cruciale pour l'étude des espaces vectoriels.

La théorie des espaces vectoriels est souvent exposée de manière purement axiomatique et formelle. Ainsi un espace vectoriel se définit comme un ensemble d'éléments (les vecteurs) muni de deux opérations internes particulières (l'addition vectorielle et la multiplication par un nombre scalaire). L'ensemble est *fermé* pour ces opérations, qui redonnent toujours des éléments de l'ensemble, c'est-à-dire des vecteurs. Cette définition formelle a l'avantage que les vecteurs peuvent être des objets très variés, comme des polynômes ou des fonctions...

En ajoutant ensuite à cette structure algébrique une opération telle que le *produit scalaire* (défini plus loin), on munit un espace vectoriel d'une mesure de distance entre vecteurs. Cette mesure permet une interprétation géométrique de l'espace vectoriel, point de vue qui se révèle souvent très intuitif et heuristique dans de nombreux problèmes.

Mais malgré son élégance théorique, la définition axiomatique n'est pas très pédagogique au premier abord. Il vaut peut-être mieux partir d'une conception plus concrète du vecteur : un vecteur est un ensemble de valeurs, ou composantes, représentant typiquement un objet ou un individu par des traits numériques. Par exemple, on peut décrire les habitants d'une ville par leur âge, revenu, niveau d'éducation, nombre d'enfants... Des traits qualitatifs (non numériques) comme le sexe, le statut marital, la profession, peuvent se traduire aisément en valeurs binaires, donc également numériques. Les traits peuvent être pondérés selon leur importance, mais ne sont pas autrement structurés entre eux.

En prenant ces valeurs comme des coordonnées dans un espace multidimensionnel, on retrouve la conception géométrique du vecteur : un point dans un espace à n dimensions (Fig. 1). Ce point correspond à un segment de droite dirigé (une flèche) à partir de l'origine des coordonnées, ce

qui est une représentation familière (bien que simpliste) des vecteurs. Le nombre de traits choisis pour décrire les individus en jeu est la dimension de l'espace vectoriel.

Cette représentation vectorielle a l'avantage de récupérer dans une certaine mesure (demandant des précautions) notre intuition de l'espace physique habituel à trois dimensions, tout en permettant un nombre de dimensions quelconque, de plusieurs milliers si nécessaire. Le caractère à la fois algébrique et géométrique de l'algèbre linéaire en font ainsi un domaine très fructueux. La théorie est maintenant très bien comprise et formalisée, et on en a tiré de nombreuses applications.

En bref, on peut voir plus ou moins intuitivement un espace vectoriel comme un espace abstrait à nombre quelconque de dimensions.

Une fois que des objets (par exemple des documents) auront été représentés par des vecteurs dans un espace vectoriel approprié, on pourra les traiter grâce aux opérations usuelles sur les vecteurs.

Les opérations de base sont l'addition vectorielle et la multiplication par un scalaire, qui sont des généralisations de l'addition et de la multiplication ordinaires. On additionne deux vecteurs en additionnant leurs composantes terme à terme (les deux vecteurs doivent avoir la même dimension), on multiplie un vecteur en multipliant chacune de ses composantes par un nombre. D'autres opérations s'en déduisent aisément, comme la soustraction vectorielle et la division par un scalaire.

Mais on cherche aussi très souvent à mesurer la ressemblance ou similitude de deux vecteurs, et on dispose pour cela d'opérations précises et faciles à calculer comme le produit scalaire. Ce dernier permet de mesurer des notions géométriques comme longueur, angle ou distance.

(dans la suite, nous noterons les vecteurs \mathbf{v} en gras)

produit scalaire

C'est la somme des produits terme à terme des composantes des deux vecteurs (les deux vecteurs doivent avoir la même dimension) :

$$\mathbf{v} \cdot \mathbf{u} = v_1 u_1 + v_2 u_2 + \dots + v_n u_n = \sum_i v_i u_i$$

Le résultat de cette opération sur deux vecteurs est un nombre (un scalaire). Celui-ci mesure la similarité relative entre deux vecteurs (il croît avec leur similarité). Pour des vecteurs à composantes binaires (0/1), c'est simplement le nombre de composantes positives communes.

Notez que le produit scalaire n'est pas une mesure absolue de similarité, car il dépend aussi de la longueur des vecteurs.

Or le produit scalaire permet de définir la *norme* d'un vecteur, c'est-à-dire sa longueur, notée $\|\mathbf{v}\|$:

$$\|\mathbf{v}\| = (\mathbf{v} \cdot \mathbf{v})^{1/2}$$

car $\|\mathbf{v}\|^2 = \mathbf{v} \cdot \mathbf{v}$ par généralisation à plus de 2 dimensions de la formule de Pythagore sur la longueur des côtés d'un triangle :

$$c^2 = a^2 + b^2$$

Si deux vecteurs ont la même longueur, leur produit scalaire est équivalent à l'angle que font les vecteurs entre eux :

angle

$$\cos \alpha = \mathbf{v} \cdot \mathbf{u} / \|\mathbf{v}\| \|\mathbf{u}\|$$

Différentes mesures utilisées en recherche d'information ne sont en fait que des variantes de cette dernière formule. Mais si les vecteurs sont de longueurs très différentes, l'angle n'est pas très significatif, et le produit scalaire est peu intuitif. Dans ce cas, on peut utiliser plutôt une distance, le plus souvent euclidienne (Fig. 2).

distance euclidienne

C'est la norme de la différence des deux vecteurs :

$$d^2 = \|\mathbf{v} - \mathbf{u}\|^2 \quad (\text{ici aussi par généralisation de la formule de Pythagore})$$

Notez que pour utiliser une telle distance, on suppose que l'espace en jeu est euclidien, c'est-à-dire homogène et vérifiant des propriétés géométriques précises comme la formule de Pythagore. Ce n'est pas toujours le cas, et on

peut alors remplacer le concept de distance par des notions moins précises de dissimilarité. Mais il faudra s'en souvenir lors des traitements ultérieurs de classification.

En effet lorsque l'on ne dispose plus d'une distance au sens strict, l'intuition géométrique devient discutable, et les classes obtenues dépendront de la mesure de similarité qui sera utilisée.

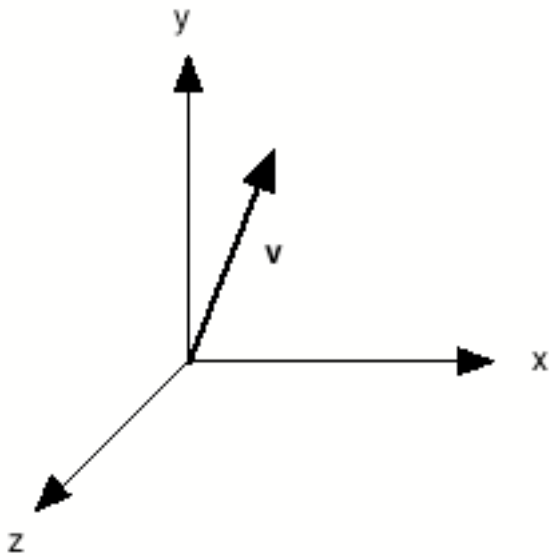


Fig. 1. Espace vectoriel

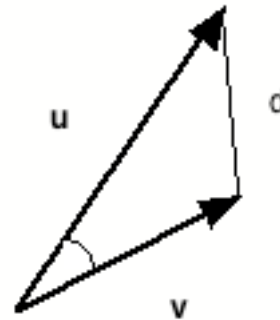


Fig. 2. Angle et distance

	mot 1	mot 2	mot 3	...	mot n
segment 1	x_{11}	x_{12}	x_{13}	...	x_{1n}
segment 2	x_{21}	x_{22}	x_{23}	...	x_{2n}
segment 3	x_{31}	x_{32}	x_{33}	...	x_{3n}
...
segment m	x_{m1}	x_{m2}	x_{m3}	...	x_{mn}

Fig. 3. Représentation vectorielle du texte

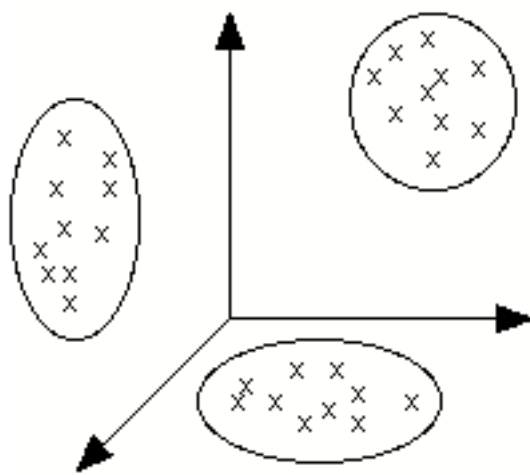


Fig. 4. Classification

vecteurs et matrices

Une autre notion importante est celle de *matrice*. Une matrice est un tableau de $m \times n$ nombres, c'est-à-dire composé de n colonnes de dimension m , ou de m rangs de dimension n . Ce peut être simplement une manière de ranger des données, mais c'est aussi un outil de calcul vectoriel, car chaque rang ou colonne de la matrice est un vecteur.

En fait, si on considère un vecteur comme une matrice à une seule colonnes (ou un seul rang), on peut généraliser le calcul vectoriel comme un cas particulier du calcul matriciel. On utilise ainsi souvent une notation uniforme des vecteurs et des matrices.

On peut donc multiplier un vecteur de dimension n par une matrice $m \times n$ pour produire un nouveau vecteur de dimension m . Cela se fait en prenant le produit scalaire du vecteur initial avec chaque vecteur rang de la matrice, ce qui donne le vecteur résultat (l'opération est en fait plus simple que sa description !). En représentant les matrices par des majuscules, un tel produit se note simplement :

$$\mathbf{u} = \mathbf{M}\mathbf{v}$$

Cette opération réalise une *transformation linéaire*, qui à tout vecteur de dimension n fait correspondre un vecteur de dimension m . Une matrice est ainsi un opérateur permettant de passer un espace vectoriel à un autre, ce qui peut être très utile, par exemple pour réduire la dimension des données (en prenant $n > m$). Mais déterminer la matrice adéquate est souvent un problème difficile, que nous ne pouvons que mentionner ici.

2. Représentation vectorielle

En représentant textes et documents dans un espace vectoriel bien choisi, on peut alors regrouper les éléments voisins de manière à faire émerger des nuages de points correspondant à des classes significatives. Par exemple des documents proches dans un tel espace traiteront en général de sujets similaires sémantiquement.

Si on accepte l'idée que cette approche est utile, il n'en reste pas moins beaucoup de questions à régler. Et il faudra être très clair sur le processus même de la représentation vectorielle si on veut pouvoir évaluer ensuite les résultats des traitements ultérieurs sur de telles représentations. Comme bien souvent, la représentation choisie va déterminer les opérations possibles et donc les résultats.

Toute représentation découle en effet de choix en partie arbitraires. Pour utiliser correctement le modèle vectoriel, il faudrait répondre aux questions suivantes :

- quels sont les éléments qu'on veut représenter ?
- quels traits pertinents choisit-on comme dimensions de l'espace ?
- comment procéder automatiquement à la représentation vectorielle des éléments en jeu ?
- y a-t-il une mesure de distance significative dans cet espace ?
- comment calculer efficacement la proximité de deux éléments ?
- en regroupant les éléments proches, quel sens peut-on attribuer aux classes ainsi formées ?

Le choix des éléments et des traits pertinents est évidemment crucial, mais les résultats dépendront aussi de l'algorithme de classification.

Il y a beaucoup de variantes attestées ou imaginables dans les réponses à ces questions. Mais un exemple typique serait le suivant :

- éléments : documents
- traits : mots importants
- distance : euclidienne

Nous verrons plus loin des algorithmes de classification courants. Pour des algorithmes de base (comme la classification par *k-moyennes*), les classes obtenues regroupent bien en général des documents sémantiquement proches, ce qui servira à des tâches pratiques.

Mais tout en restant dans le cadre vectoriel, il existe d'autres possibilités plus complexes, que nous aurons l'occasion de mentionner et de discuter par la suite.

3. Processus de traitement

Le traitement de textes et documents selon le modèle vectoriel suit généralement des étapes typiques : vectorisation, classification, interprétation et utilisation (Salton & McGill 83). Chacune de ces étapes devra répondre à son tour aux questions de la section précédente. Mais c'est l'étape de vectorisation (c'est-à-dire de représentation d'un texte par un ou plusieurs vecteurs) qui demande de faire le choix fondamental des éléments retenus et des traits représentatifs. Les choix dépendent bien sûr des tâches à effectuer (indexation ou étude terminologique par exemple), et nous allons passer en revue les principales possibilités.

Choix des éléments

Pour un document court comme une page Web ou un courrier électronique, on peut représenter tout le texte par un vecteur. Pour un document plus long comme un article scientifique, on peut ne représenter qu'une partie significative (le résumé, l'introduction, la première page, les titres et sous-titres... ou une combinaison des éléments précédents) ou bien encore représenter tout le texte par un seul vecteur.

Pour des documents plus longs comme un livre ou un manuel, l'ensemble n'est généralement pas suffisamment homogène pour donner lieu à une représentation unique significative. On y perdrait en tous cas beaucoup d'information, ce qui ne se justifierait que pour un classement grossier. Il vaut mieux découper le document en unités plus petites : chapitre, page, paragraphe... Chacun de ces segments de texte sera alors représenté par un vecteur de traits lexicaux, et l'ensemble du texte peut se représenter comme une matrice (Fig. 3). C'est la comparaison de ces vecteurs qui pourra renseigner sur l'ensemble du document.

Choix des traits descriptifs

Les traits pertinents à retenir pour la description des éléments précédents sont soumis à des considérations contradictoires. Ils doivent être représentatifs et discriminants mais faciles à extraire. Il est préférable que les traits ne soient pas trop nombreux, car ils fixent la dimension de l'espace vectoriel en jeu,

mais il est aussi souhaitable que les prétraitements à faire pour réduire leur nombre ne soient pas trop lourds.

On se sert souvent comme traits des mots présents dans le documents. Mais on commence par éliminer les mots de fonction (comme articles, prépositions, particules diverses) qui sont très fréquents et peu discriminants. Ensuite on ne retient que les mots pleins (surtout noms et verbes) ayant une fréquence suffisante dans le document. Cela revient à ne garder que les mots de fréquence moyenne, ni trop haute ni trop basse.

On peut aussi utiliser des mesures plus élaborées pour ne retenir que les mots les plus discriminants entre documents dans l'ensemble d'un corpus. Une mesure assez simple consiste à diviser la fréquence d'un mot par le nombre de documents où il apparaît (car un mot présent partout sera peu discriminant). Mais on pourrait en citer d'autres encore.

Il est aussi utile dans les langues à morphologie variable (par exemple noms dans les langues slaves ou verbes dans les langues romanes) de *lemmatiser* les formes, c'est à dire de les ramener à une racine commune avant de les compter. Ainsi on pourra ramener à une même forme *soudaient*, *soudront*, *soudé* ainsi éventuellement que *soudure*, *soudeur*. Il existe pour cela des programmes d'analyse morphologique, mais l'emploi d'une liste de suffixes ou une troncature des lettres finales marchent aussi très bien en pratique.

Un autre choix à faire est de ne noter que la présence ou l'absence d'un mot dans un segment de texte (donnant ainsi des vecteurs binaires) ou bien de retenir la fréquence du mot dans le segment (ou des mesures plus élaborées) de manière à pondérer le trait descriptif.

Représentation mot / document

Il existe aussi une approche duale : représentation des mots ou termes dans l'espace des documents, et classification des mots dans ce contexte. Les éléments à classer sont alors les mots, et les traits descriptifs sont des documents (ou des segments de texte). Cela permet de créer des classes sémantiques de mots, notamment pour créer un dictionnaire de synonymes ou plus généralement un *thésaurus* de termes apparentés (Salton 72) (Charniak 93, chap. 9).

Un thésaurus est utile pour élargir une recherche de documents en ajoutant à la requête d'origine des mots auxquels l'utilisateur n'a pas pensé, mais qui

seraient pertinents. Mais il y a d'autres usages encore (aide à la rédaction, traduction automatique) pour ce genre de répertoire.

Une fois ces divers choix faits, on va employer une méthode de classification pour regrouper les vecteurs similaires et faire ainsi émerger une structure sémantique sur l'ensemble des documents traités. Ici aussi, il y a des méthodes diverses que nous allons voir plus loin.

4. Problèmes et variantes

Cependant le problème des méthodes lexicales est qu'on aboutit souvent à un très grand nombre de traits : plusieurs centaines ou plusieurs milliers. Des dimensions aussi élevées posent de sérieux problèmes de représentation, de calcul et d'interprétation. C'est ce qu'on a pu appeler la "malédiction dimensionnelle" (*curse of dimensionality*). On a alors affaire à des vecteurs très creux (*sparse*) avec une grande majorité de composantes nulles. Il est important d'essayer de réduire la dimension de l'espace en diminuant le plus possible le nombre de traits pertinents, ou en trouvant une transformation vectorielle appropriée.

On a proposé pour cela des techniques diverses, le plus souvent d'inspiration probabiliste. Par exemple on peut employer une matrice aléatoire pour multiplier les vecteurs initiaux et les transformer en vecteurs de dimension moindre (Ritter & Kohonen 89) (Kohonen 98). Si cette transformation respecte plus ou moins la distribution initiale en la compressant, cela peut être très efficace. Sinon on risque de perdre beaucoup d'information.

On peut utiliser la théorie de l'information ou bien encore l'analyse factorielle (Bouroche & Saporta 80) pour calculer la pertinence des traits lexicaux pour les traitements ultérieurs, de manière à ne retenir qu'un nombre réduit de traits. Il y a alors un compromis à trouver entre la complexité de ces prétraitements et les économies de temps et de mémoire qu'ils permettent par la suite. Il faut aussi essayer d'optimiser représentations et traitements.

Ensuite nous avons employé jusqu'ici la notion de *mot* de manière relativement vague. Mais il faudrait préciser si on parle de formes fléchies, de la forme de base, éventuellement d'une forme plus abstraite commune à toute une famille de mots... Il y a aussi le cas des mots composés qui peuvent

représenter un concept bien spécifique. On emploie donc souvent plutôt la notion de *terme* pour désigner un "mot" plus abstrait, sémantiquement important dans un domaine donné, qui fera donc un meilleur trait descriptif qu'une occurrence de surface.

Un autre problème est que dans certaines langues, il n'est pas évident de détecter les mots, dont les frontières ne sont pas clairement marquées dans l'écriture : c'est par exemple le cas dans des langues aussi importantes que le chinois, le japonais ou l'arabe. On peut alors utiliser des unités plus fines (comme les lettres ou les caractères chinois) et compter des séquences d'unités (des *n-grammes*). C'est moins intuitif, mais les résultats sont théoriquement justifiés et tout à fait acceptables en pratique.

Enfin il ne faut pas oublier que la classification n'aura de sens que par rapport à une tâche donnée. Il faudra évaluer les résultats soit par les performances dans la réalisation de la tâche (recherche de documents par exemple), soit en prenant l'avis d'un expert humain (par exemple en terminologie). Cette étape d'interprétation et d'évaluation est inévitable.

5. Méthodes de classification

La classification est une étape essentielle du processus de traitement, car elle assemble les éléments similaires en groupes homogènes, correspondant à des régions de l'espace vectoriel (Fig. 4). La partition en classes des données permet de réduire la taille (et la dimension) de l'ensemble initial, tout en faisant apparaître une organisation significative (Anderberg 73) (Everitt 80) (Roux 86) (Van Cutsem 94) ; pour une introduction voir (Lechevallier 97). Ainsi la classification de documents servira à révéler les thèmes majeurs du corpus traité.

Précisons que nous employons ici le terme de "classification" au sens statistique de regroupement d'éléments similaires (*clustering*), sans disposer d'exemples d'éléments déjà classés et étiquetés. Autrement dit, il s'agit de classification non supervisée, sans connaissances *a priori*. Dans ce cas, classer signifie suivre la distribution des données dans l'espace de façon à la découper au mieux en un ensemble de groupes.

Il existe pour cela deux grand types de classification : les méthodes hiérarchiques et non-hiérarchiques. Les méthodes non-hiérarchiques donnent

directement une classification à un seul niveau de partition, et sont plus simples (et plus efficaces pour de grands ensembles de données). Nous traiterons ici essentiellement de classification non-hiérarchique comme l'algorithme des k -moyennes.

Mais il convient de mentionner les méthodes hiérarchiques, fournissant un arbre binaire de classes emboîtées. Il existe divers algorithmes ascendants fonctionnant par regroupement successif de classes à chaque niveau. Mais les différents niveaux de la hiérarchie sont souvent difficile à interpréter, notamment pour les niveaux intermédiaires.

Rappelons aussi que toute classification non supervisée suppose que l'on dispose d'une mesure cohérente de distance ou de similarité entre éléments (que cette mesure soit explicite, ou implicite dans le choix d'une méthode particulière). D'autre part les techniques de classification sont des algorithmes heuristiques, donnant de "bonnes" classes mais sans garantir l'optimalité de la partition obtenue (il est pratiquement infaisable d'examiner toutes les combinaisons possibles).

Il existe un certain nombre de méthodes de classification non-hiérarchique, et on se contentera d'en examiner deux ou trois exemples pour en donner une idée. Les algorithmes fonctionnent le plus souvent par amélioration itérative d'une partition initiale. On peut distinguer des techniques statistiques classiques comme la méthode des k -moyennes (*k-means*) et ses variantes, et des méthodes neuronales comme les réseaux à apprentissage compétitif. Mais nous détaillerons un peu plus les réseaux de neurones, qui restent moins connus dans ce domaine.

Méthode des k -moyennes

On part de k centres arbitraires autour desquels on regroupe les éléments les plus proches de ces centres (Lechevallier 97). On calcule le centre de gravité (c'est-à-dire la moyenne des vecteurs) de chaque classe au fur et à mesure du regroupement des éléments, et on prend ces points comme nouveaux centres. On effectue une nouvelle classification de l'ensemble, et ainsi de suite, jusqu'à ce que la dispersion des membres de chaque classe soit minimale. En effet les classes deviennent plus compactes à chaque nouvelle partition, et l'algorithme converge rapidement.

Il existe plusieurs variantes très voisines de cette méthode de base. L'algorithme des "centres mobiles" ne recalcule le centre de gravité d'une

classe qu'après regroupement de tous les nouveaux membres. La méthode des "nuées dynamiques" utilise un noyau de plusieurs éléments réels plutôt qu'un centre de gravité calculé pour chaque classe.

Ces méthodes sont rapides et efficaces, mais elles demandent de fixer le nombre k de classes au départ, et le résultat dépend des centres (ou noyaux) initiaux. Il est souvent nécessaire de faire varier le nombre de classes et les centres de départ. Lorsqu'on a une idée préalable de la répartition des classes, on améliorera la classification en choisissant au mieux les k centres initiaux. On peut aussi essayer de repérer les sous-ensembles d'éléments qui se retrouvent toujours ensemble quelque que soit la classification ("formes fortes") et en faire des points de départ.

Enfin les centres finaux (ou "centroïdes") sont représentatifs des classes obtenues : on peut considérer un centre comme le prototype de sa classe. Ce vecteur pourra représenter avantageusement sa classe pour des traitements ultérieurs.

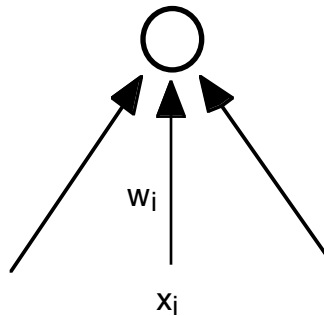


Fig. 5. Neurone élémentaire

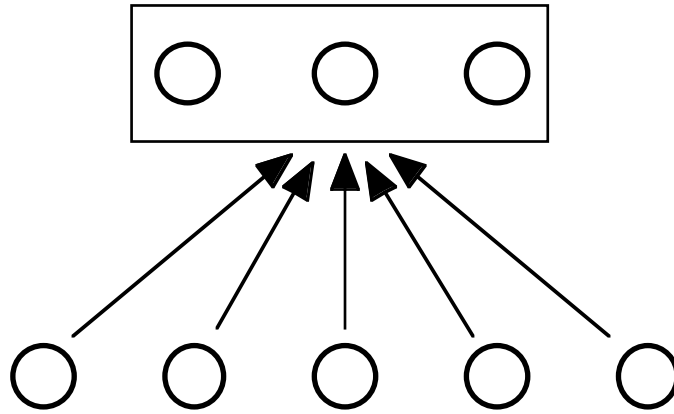


Fig. 6. Réseau compétitif

(on n'a représenté les connexions que vers un seul neurone de sortie)

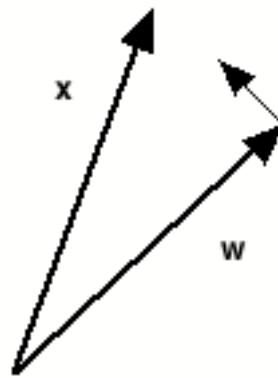


Fig. 7. Règle d'apprentissage

Réseaux compétitifs

C'est une famille de modèles neuronaux qui permet de faire de la classification non supervisée (Grossberg 87) (Rumelhart & Zipser 91) (Kohonen 97). L'idée de base est similaire à celle des k-moyennes : on part de classes initiales plus ou moins arbitraires, et on améliore progressivement la classification en ajustant itérativement le vecteur prototype de chaque classe.

Par analogie avec la neurobiologie, un réseau neuronal se compose de neurones formels connectés entre eux, et à chaque connexion est associé un poids modulant la transmission d'activité sur cette ligne. C'est la transmission d'activité dans le réseau qui réalise une tâche donnée, et les poids sont modifiables de manière à assurer les résultats souhaités. Pour une présentation générale des méthodes neuronales, voir (Rumelhart & McClelland 86) (Hertz et al. 91) (Hérault & Jutten 94) (Jodouin 94) (Thiria et al. 97).

Le type de neurone le plus simple calcule la somme linéaire de ses entrées modulées par les poids des connexions (Fig. 5). Ceci n'est autre que le produit scalaire du vecteur d'entrée et du vecteur de poids, et ce neurone mesure donc la similarité entre les deux vecteurs. On retrouve les concepts de l'algèbre linéaire, et on va voir comment les mettre en pratique pour effectuer une classification.

activation du neurone linéaire : $y = \sum_i w_i x_i = \mathbf{w} \cdot \mathbf{x}$

(\mathbf{w} est le vecteur de poids, et \mathbf{x} est le vecteur d'entrée)

Dans un réseau compétitif, on présente des vecteurs d'entrée (représentant les éléments à classer) à une couche de neurones (Fig. 6). Chaque neurone de sortie calcule le produit scalaire entre son vecteur de poids et le vecteur d'entrée. Le neurone dont le vecteur de poids est le plus proche du vecteur d'entrée donne la réponse la plus forte, et on considère alors qu'il représente la classe de l'élément d'entrée.

En principe (pour suivre l'analogie biologique) une compétition par inhibition entre les neurones déciderait du neurone gagnant, mais il suffit en pratique de choisir le neurone ayant l'activation la plus élevée. En fait, pour éviter les normalisations et accélérer les calculs, on préfère souvent choisir

plutôt le neurone présentant la distance minimale entre vecteur d'entrée et vecteur de poids :

$$\min d^2 = \|\mathbf{x} - \mathbf{w}\|^2$$

Puis on modifie itérativement les poids des connexions de manière à rapprocher le vecteur de poids de chaque neurone gagnant vers le vecteur d'entrée (Fig. 7). On peut arrêter le processus quand les modifications deviennent trop petites. La règle d'adaptation est la suivante :

$$\Delta \mathbf{w} = \varepsilon (\mathbf{x} - \mathbf{w}) \quad (\varepsilon \text{ étant un pas d'apprentissage } 0 < \varepsilon < 1).$$

La procédure a donc pour effet que chaque neurone se spécialise progressivement dans la reconnaissance d'une classe et devient représentatif de cette classe. On reproduit ainsi au mieux la répartition des données dans l'espace. De manière comparable à la méthode des k-moyennes, on peut considérer le vecteur de poids final d'un neurone comme le prototype de la classe représentée.

Il existe plusieurs variantes à cet algorithme de base : la plus simple est sans doute la méthode VQ (*Vector Quantization*), et sa variante semi-supervisée LVQ (*Learning Vector Quantization*) qui permet d'affiner les frontières entre classes (Kohonen 87).

Les réseaux ART (*Adaptive Resonance Theory*) créent dynamiquement des classes supplémentaires quand de nouvelles entrées sont trop différentes des classes actuelles, de façon à éviter les problèmes d'instabilité de l'apprentissage (Carpenter & Grossberg 88). Ils sont intéressants quand on risque d'avoir en continu de nouvelles entrées de forme imprévue, ce qui peut être le cas des textes (Meunier & Nault 95 ; 97) (Memmi et al. 98).

Les cartes de Kohonen (ou *Self-Organizing Maps*) présentent l'intérêt d'ordonner topologiquement les classes obtenues sous forme d'une carte, généralement sur un plan (Kohonen 97). Cette représentation visuelle est très agréable, comparable aux résultats d'une analyse factorielle (mais le traitement est ici non-linéaire, donc potentiellement plus puissant). On obtient ainsi des cartes sémantiques de documents (Ritter & Kohonen 89) (Honkela 97) (Kohonen 98).

Un avantage de ces méthodes neuronales est qu'il n'est pas nécessaire de fixer le nombre de classes *a priori*, ce qui peut se révéler fort utile, et l'analogie biologique reste une source fertile de variantes intéressantes. Mais ces algorithmes sont plus difficiles à caractériser théoriquement que les méthodes statistiques classiques.

Noyau caractéristique : nous proposons de nommer ainsi une notion qui nous sera utile par la suite. Le "noyau caractéristique" d'une classe est le sous-ensemble de traits caractérisant le centroïde de la classe (c'est-à-dire les traits communs à tous les vecteurs membres). Pour des traits binaires, ce noyau est simplement l'intersection des traits positifs de tous les membres de la classe. Pour des valeurs continues, il faudra prendre un seuil pour décider quels sont les traits de valeur suffisamment élevée pour être significative.

Plus concrètement, pour une classe de documents ce noyau sera la liste des mots pleins communs à tous les documents de la classe. Ces mots sont évidemment représentatifs de la classe, et serviront de base à des traitements ultérieurs. Ce sont par exemple de bons candidats pour établir une liste des termes essentiels du domaine traité.

6. Applications principales

Le modèle vectoriel s'est révélé tout à fait efficace pour de nombreuses applications. Certaines sont devenues des méthodes standard, d'autres restent des domaines de recherche. Mais on peut remarquer la variété des applications d'un modèle somme toute assez simple. La diversité des travaux est telle que nous ne présenterons ici qu'un aperçu des principales directions.

Indexation et recherche

C'est l'application la plus évidente et la plus courante des méthodes vectorielles. Après vectorisation des documents par les techniques exposées plus haut, on obtient des classes de documents proches dans l'espace vectoriel. Le "noyau caractéristique" des mots communs à une classe de documents peut servir d'index, puisqu'il caractérise la classe.

La recherche des documents pertinents se fera alors à partir d'une requête composée de mots-clefs. On considère que la requête est un vecteur se situant

dans le même espace que les documents, on recherche la classe dont le centroïde est le plus proche de la requête, et on ramène les documents de cette classe. Le système SMART est un des exemples les plus connus (Salton 71) (Salton & McGill 83, chap. 4).

L'utilisation d'un thésaurus permet d'enrichir la requête initiale avec des termes apparentés pour élargir la recherche. Il est également possible d'autoriser des requêtes plus structurées (en format booléen ou en langage naturel par exemple), qui seront traduites en un ensemble de mots-clefs.

Il existe de nombreuses variantes, mais la méthode reste fondamentalement la même. Les résultats sont généralement excellents en rapidité de calcul, et acceptables en performances sémantiques. On mesure ces performances par le taux de *rappel* (pourcentage de documents retrouvés parmi tous les documents pertinents) et le taux de *précision* (pourcentage de documents pertinents parmi les documents ramenés). En d'autres termes, il faut essayer de minimiser à la fois le silence et le bruit dans les résultats.

C'est en gros ainsi que fonctionnent les moteurs d'indexation et de recherche sur le Web (Leloup 97) bien que leur langage de requêtes présente souvent un format booléen en surface. Mais le Web pose des problèmes particuliers. Etant donné la taille des corpus en jeu (des millions de documents en évolution constante), les systèmes actuels ont encore des performances médiocres en précision, et le taux de rappel n'est pas accessible. Et il reste beaucoup à faire pour l'aide à l'utilisateur.

Filtrage et résumé

Une autre application assez proche de la précédente consiste à utiliser la classification non pas pour rechercher des documents pertinents, mais pour exclure des messages non pertinents. C'est utile pour le courrier électronique par exemple, afin d'éviter des messages non désirés. Le système de filtrage ne laissera passer que des textes proches des centres d'intérêt de l'utilisateur. Les navigateurs les plus récents (Netscape, Explorer) offrent des embryons élémentaires de cette idée.

Les filtres peuvent être explicitement initialisés par l'utilisateur, ou bien déduits de son comportement. Le système notera ce que l'utilisateur accepte ou rejette, et en tirera des patrons positifs ou négatifs. Ce sont toujours des vecteurs, avec lesquels seront comparés tout nouveau message électronique.

De la même façon, on peut aider à la navigation sur le Web, en observant le comportement de l'utilisateur. Après un temps d'apprentissage, le système proposera à l'utilisateur les pages et les liens les plus proches de ses centres d'intérêt constatés précédemment. L'apprentissage peut d'ailleurs se faire en continu au fur et à mesure de l'évolution du comportement de navigation.

De plus, on peut décrire sommairement le contenu des textes recommandés ou déconseillés par un tel système. En effet les mots qui ont servi à accepter ou à rejeter un document constituent une description grossière mais significative. Eventuellement, un résumé plus élaboré pourra être fourni à la demande (effet de "zoom").

Extraction terminologique

Dans les grandes organisations industrielles ou bureaucratiques, il est important de maintenir une terminologie homogène et uniforme (ne serait ce que pour la recherche documentaire). L'approche vectorielle peut aider à cela. Après classification de documents ou de segments de documents, on recueille le noyau caractéristique des classes obtenues. Ces noyaux lexicaux donnent de précieuses indications sur les thèmes principaux et les régularités terminologiques du corpus traité (Meunier & Nault 97) (Memmi & Meunier 2000). En effet les noyaux représentent les ensembles les plus fréquents de cooccurrences croisées, qui ont servi de base à la classification.

Il s'agit là d'une aide à l'extraction de connaissances, et non d'un processus entièrement automatique. Il reste nécessaire de faire appel à un expert humain pour valider et exploiter les résultats. Mais il y a un gain de temps certain par rapport à une approche manuelle, et on peut développer des programmes de post-traitement pour mieux révéler et mettre en forme les relations entre termes du corpus.

Construction de thésaurus

Il est souvent utile de disposer de dictionnaires de synonymes et de termes apparentés. Cela permet notamment d'élargir une recherche en ajoutant à une requête des mots auxquels l'utilisateur n'aura pas pensé spontanément (par exemple associer *bateau* et *voilier* à *navire*). Cela sert aussi à normaliser une requête en remplaçant les mots de l'utilisateur par des mots-clefs voisins, ou encore à aider à la rédaction de documents. Un tel répertoire ou *thésaurus* peut

se réaliser en classant les mots dans l'espace des textes où on les trouve (Salton 72).

Il faut pour cela découper le corpus de textes en documents ou fragments homogènes, qui vont constituer les dimensions de l'espace de référence. Les mots qui se retrouvent (en fréquence suffisante) dans les mêmes régions de cet espace sont probablement apparentés, puisqu'ils sont présents dans les mêmes documents. On peut faire varier la taille des textes de référence, c'est-à-dire la "fenêtre" dans laquelle on constate la cooccurrences des mots. Selon la taille de cette fenêtre, on passera de liens sémantiques généraux à des relations linguistiques plus fines (Grefenstette 94).

Autres directions

Il existe aussi des directions de recherche plus avancées, dont les résultats, bien que prometteurs, restent plus incertains. On peut en mentionner brièvement un certain nombre :

- recherches linguistiques

L'approche vectorielle constitue une méthode d'investigation de la structure sémantique d'un corpus de texte. L'extraction de termes par exemple n'est pas seulement un problème pratique, mais pose aussi des questions sur le contenu sémantique des textes.

- extraction de connaissances

En poursuivant les travaux sur l'extraction de termes, on peut espérer extraire les thèmes essentiels d'un texte, et par là même des connaissances sur le domaine décrit. Cette direction de recherche, à peine esquissée actuellement, se heurte peut-être à des problèmes de fond.

- informations multimédia

Le modèle vectoriel peut aussi s'appliquer à des documents contenant des images, du son, de la vidéo... Le problème est alors de trouver une représentation adéquate pour chaque format d'information (comment décrire une image ?) et de relier entre eux ces divers types de représentation (mais le texte restera probablement la forme de base).

- aide à l'utilisation du Web

La recherche et la navigation sur le Web est évidemment un domaine d'application majeur pour ces différents travaux. Une direction actuelle consiste à personnaliser la navigation en recueillant les thèmes d'intérêt habituels de l'utilisateur pour le guider ensuite dans ses recherches. Les profils de consultation ainsi établis constituent une expertise explicite et réutilisable.

- interfaces graphiques

Le modèle vectoriel se prête particulièrement bien à des représentations graphiques intuitives d'un ensemble de documents, termes ou thèmes... Il faut pour cela projeter l'espace en jeu sur une carte (généralement en deux dimensions) en conservant le plus possible d'information pertinente. Le problème s'apparente à l'analyse factorielle, mais il relève aussi de l'ergonomie des interfaces.

Il ne serait pas très difficile d'allonger cette liste d'applications potentielles et de thèmes de recherche, mais il reste aussi des problèmes fondamentaux.

7. Remarques théoriques

Le modèle vectoriel suscite plusieurs types de réflexions. Certaines portent sur le rapport à la théorie mathématique, d'autres sur l'interprétation linguistique des résultats de cette approche.

Remarques mathématiques

La rigueur formelle de la théorie mathématique sous-jacente (algèbre linéaire et espaces vectoriels) ne doit pas trop faire illusion sur l'utilisation pratique du modèle. En effet les représentations réellement employées ne constituent pas rigoureusement des espaces vectoriels.

Un espace vectoriel suppose que les axes de coordonnées soient indépendants et de nombre minimal (c'est la définition même de la *base* d'un espace). Ce n'est évidemment pas le cas des traits lexicaux servant à représenter les textes. Le choix des traits est relativement arbitraire, et il y a des dépendances statistiques entre mots ou termes, dépendances qu'on s'abstient généralement de calculer. Les notions de distance ou d'angle doivent

aussi être utilisées avec précaution (elles supposent en toute rigueur des axes orthonormés).

Ceci dit, on se livre en analyse factorielle à de pareilles approximations sans trop de scrupules. En Analyse en Composantes Principales (ACP) par exemple, on y représente les individus à comparer dans un espace vectoriel dont les dimensions sont corrélées, dans le but de trouver de nouveaux axes de représentation qui soient moins nombreux et indépendants. Les résultats de l'analyse justifient largement ce choix initial théoriquement discutable.

En somme, il y a un certain décalage entre la rigueur mathématique des méthodes employées, et le caractère assez approximatif des données traitées. Il y a donc en fait une bonne part d'expérimentation inhérente à l'approche vectorielle.

Ensuite l'intuition géométrique acquise dans l'expérience d'un espace habituel à trois dimensions peut se révéler trompeuse dans des espaces à grand nombre de dimensions (Hérault & Guérin-Dugué 97). Par exemple le volume de différentes figures augmente différemment lorsque la dimension de l'espace augmente.

Questions linguistiques

On doit aussi se demander quel est vraiment le sens linguistique des représentations et opérations utilisées dans le modèle vectoriel. Nous avons supposé dès le départ que la proximité de documents (ou de mots) dans un espace bien choisi révélait une ressemblance sémantique, et les opérations de classification sont basées sur cette idée. On arrive bien à effectuer ainsi des tâches comme la recherche documentaire ou l'extraction terminologique, mais cela mériterait une réflexion plus poussée dans le cadre général du TALN.

On remarquera notamment que cette proximité sémantique est essentiellement *lexicale*. Dans cette optique, deux documents sont considérés comme voisins s'ils contiennent une proportion importantes de mots communs. C'est en fait une supposition simpliste. Des documents pourraient traiter de thèmes similaires en ayant très peu de mots pleins en commun. On a vu que l'emploi d'un thésaurus listant des termes apparentés permet d'élargir le champ de la proximité sémantique, mais l'approche restera lexicale.

De plus la représentation vectorielle d'un document est *globale* : c'est une liste non structurée de termes présents dans le texte. On perd les relations locales (et notamment les relations syntaxiques). On peut alors travailler avec

des documents ou segments de texte plus courts, pour augmenter la précision locale. Mais cela demanderait aussi des traitements plus longs.

En poursuivant dans cette direction, on pourrait vouloir recourir aux méthodes formelles élaborées en linguistique computationnelle pour l'analyse de phrases et l'analyse de textes. Ces méthodes permettent d'obtenir des représentations sémantiques plus structurées. Mais elles sont aussi beaucoup plus coûteuses et fragiles à mettre en oeuvre, et donc peu appropriées au traitement de gros corpus.

Un compromis envisageable consisterait à utiliser à la fois des traits lexicaux et des traits plus complexes portant sur des expressions et structures syntaxiques simples. On rejoindrait ainsi certains travaux actuels en terminologie (voir Frath 97, chap 1) et en analyse superficielle (*shallow parsing*). Mais dans le domaine de la recherche documentaire, cela reste à notre avis une direction de travail dont l'efficacité n'est pas encore totalement prouvée.

Conclusions

Nous avons voulu ici présenter les notions essentielles nécessaires pour une bonne compréhension du modèle vectoriel et de ses applications au traitement de documents. Nous espérons avoir montré l'utilité et la diversité des applications d'une approche fondamentalement très homogène. Mais nous pensons aussi que beaucoup reste à faire dans un domaine qui ne peut que se développer pour des raisons socio-économiques évidentes (il suffit de mentionner l'extraordinaire croissance d'Internet et des Intranets).

On pourrait alors se contenter de traiter les problèmes techniques au fur et à mesure des besoins, et la recherche d'information en particulier a souvent fait preuve d'un empirisme notable. Il est possible de voir tout ce domaine comme relevant d'une ingénierie qui ne s'embarrasse pas trop de considérations sémantiques, et l'efficacité d'une telle approche est manifeste. Elle est probablement même inévitable lorsque les questions d'efficacité sont primordiales comme dans le traitement de très gros corpus.

Cependant ce domaine touche aussi inévitablement à des disciplines comme la linguistique, la sémiologie, la psychologie cognitive, l'ergonomie... En effet c'est l'utilisateur qui est l'ultime juge de la validité des techniques employées,

et on n'a pas encore suffisamment travaillé à notre avis sur les représentations qu'il se fait des objets et opérations en jeu. Le rôle des interfaces graphiques, qui est très important en pratique, demanderait une réflexion plus poussée.

Ensuite on ne peut pas oublier que l'objet même des traitements est constitué de textes en langage naturel (éventuellement accompagné d'autres formes de représentation). Les choix de base préliminaires aux traitements sont, explicitement ou non, des choix linguistiques demandant un minimum de compétence en ce domaine (et posant des problèmes de fond). Quant aux résultats obtenus, ils sont souvent fort intéressants pour l'étude du langage lui-même, en terminologie par exemple.

Il reste donc à replacer l'approche vectorielle dans le cadre général du traitement du langage. Pourtant une comparaison détaillée avec le TALN syntaxique ne serait sans doute pas très pertinente, car cette approche numérique poursuit des buts différents. Il vaut mieux chercher à voir quelles sont les conséquences de ces objectifs. Le modèle vectoriel considère un texte dans son ensemble et cherche à inférer du texte une structure sémantique implicite. Par delà les variantes, c'est généralement un réseau de termes correspondant aux thèmes traités dans le texte.

Cette organisation sémantique n'est pas tirée directement de l'examen des phrases du texte, mais on a vu qu'elle découle en fin de compte des cooccurrences lexicales. Les représentations internes étant numériques plutôt que symboliques, les traitements sont rapides, très résistants au bruit, et fournissent naturellement des résultats pondérés si nécessaire. Mais on n'utilise pas l'information syntaxique (du moins dans la version de base du modèle).

En résumé, l'approche vectorielle est globale et floue. Elle fait appel à une méthodologie numérique encore peu familière en TALN, et elle néglige une partie de l'information linguistique fine contenue dans un texte. Mais elle est robuste et rapide, et se révèle à la fois efficace et fructueuse dans des applications de grande importance pour notre société de l'information.

Remerciements : nous avons bénéficié de nombreuses discussions sur l'algèbre linéaire avec A. Birebent et J.L. Dorier de l'équipe Didactique des Mathématiques au laboratoire Leibniz. Mais nous sommes bien sûr responsable des interprétations que nous en avons tirées.

Références

- Abeillé A. & Blache Ph. (1997) Etat de l'art : la syntaxe, *T.AL*. 38 (2).
- Anderberg M.R. (1973) *Cluster Analysis for Applications*, Academic Press.
- Bourbaki N. (1947) *Eléments de Mathématiques*, livre II chap. 2 : Algèbre linéaire, Hermann.
- Bouroche J.M. & Saporta G. (1980) *L'Analyse des Données*, Que sais-je n° 1854, PUF.
- Carpenter G.A. & Grossberg S. (1988) The ART of adaptive pattern recognition by a self-organizing neural network, *IEEE Transactions on Neural Networks*, March 1988, p. 77-88.
- Charniak E. (1993) *Statistical Language Learning*, MIT Press.
- Dorier J.L. (1995) A general outline of the genesis of vector space theory, *Historia Mathematica* 22, p. 227-261.
- Everitt B. (1980) *Cluster Analysis*, Halsted.
- Frath P. (1997) *Sémantique, Référence et Acquisition Automatique de Connaissances à partir de Textes*, Thèse Université des Sciences Humaines de Strasbourg.
- Grefenstette G. (1994) *Explorations in Automatic Thesaurus Discovery*, Kluwer.
- Grossberg S. (1987) Competitive learning: from interactive activation to adaptive resonance, *Cognitive Science* 11, p. 23-63.
- Habert B., Nazarenko A. & Salem A. (1997) *Les Linguistiques de Corpus*, Colin.
- Halmos P.R. (1974) *Finite-Dimensional Vector Spaces*, Springer Verlag.
- Hérault J. & Guérin-Dugué A. (1997) Analyse de données multidimensionnelles par réseaux de neurones auto-organisés, chap. 9 in Thiria et alii, *Statistique et Méthodes Neuronales*, Dunod.
- Hérault J. & Jutten C. (1994) *Réseaux Neuronaux et Traitement du Signal*, Hermès.
- Hertz J., Krogh A. & Palmer R.G. (1991) *Introduction to the Theory of Neural Computation*, Addison Wesley.
- Honkela T. (1997) *Self-Organizing Maps in Natural Language Processing*, Ph.D. Thesis, Helsinki University of Technology.

- Jodouin J.F. (1994) *Les Réseaux Neuromimétiques*, Hermès.
- Johnson L.W., Riess R.D. & Arnold J.T. (1998) *Introduction to Linear Algebra*, Addison-Wesley.
- Jolliffe I.T. (1986) *Principal Component Analysis*, Springer Verlag.
- Jordan M.I. (1986) An introduction to linear algebra in parallel distributed processing, chap. 9 in Rumelhart & McClelland, *Parallel Distributed Processing*, MIT Press.
- Kohonen T. (1987) *Self-Organization and Associative Memory*, Springer Verlag.
- Kohonen T. (1997) *Self-Organizing Maps*, Springer Verlag.
- Kohonen T. (1998) Self-organization of very large document collections: state of the art, *Proc. of ICANN'98*, London.
- Lebart L. & Salem A. (1994) *Statistique Textuelle*, Dunod.
- Lechevallier Y. (1997) Classification non supervisée, chap. 10 in Thiria et alii, *Statistique et Méthodes Neuronales*, Dunod.
- Leloup C. (1997) *Moteurs d'Indexation et de Recherche*, Eyrolles.
- Manning C.D. & Schütze H. (1999) *Foundations of Statistical Natural Language Processing*, MIT Press.
- Memmi D., Gabi Kh. & Meunier J.G. (1998) Dynamical knowledge extraction from texts by ART networks, *Proc. of NEURAP'98*, Marseille.
- Memmi D. & Meunier J.G. (2000) Using competitive networks for text mining, *Proc. of NC'2000*, Berlin.
- Meunier J.G. & Nault G. (1995) Modèles connexionnistes et traitement de l'information textuelle : le modèle ART de Grossberg, Cahiers de Recherche LANCI 95.6, UQAM, Montréal.
- Meunier J.G. & Nault G. (1997) Approche connexionniste au problème de l'extraction de connaissances terminologiques à partir de textes, in "Les Techniques d'Intelligence Artificielle Appliquées aux Technologies de l'Information", Lepage R. & Noumeir R. (eds), *Les Cahiers Scientifiques de l'ACFAS*, No. 90, p. 62-76.
- Ritter H. & Kohonen T. (1989) Self-organizing semantic maps, *Biological Cybernetics* 61 (4), p. 241-254.
- Roux M. (1986) *Algorithmes de Classification*, Masson.
- Rumelhart D.E. & McClelland J.L. (eds.) (1986) *Parallel Distributed Processing*, MIT Press.

- Rumelhart D.E. & Zipser D. (1985) Feature discovery by competitive learning, *Cognitive Science* 9, p. 75-112.
- Sabah G. (1990) *L'Intelligence Artificielle et le Langage*, Hermès.
- Salton G.(ed.) (1971) *The SMART Retrieval System - Experiments in Automatic Document Processing*, Englewood Cliffs.
- Salton G. (1972) Experiments in Automatic Thesaurus Construction for Information Retrieval, *Information Processing* 71, p. 115-123, North-Holland.
- Salton G. & McGill M. (1983) *Introduction to Modern Information Retrieval*, McGraw-Hill.
- Salton G. & Buckley C. (1994) Automatic structuring and retrieval of large text files, *Communications of the ACM* 37 (2), p. 97-107.
- Seffah A. & Meunier J.G. (1996) ALADIN: an integrated object-oriented environment for computer assisted text analysis, Cahiers de Recherche LANCI 96.1, UQAM, Montréal.
- Strang G. (1976) *Linear Algebra and its Applications*, Academic Press.
- T.A.L. (1995) Traitements Probabilistes et Corpus, n° spécial, *T.A.L.* 36 (1-2).
- Thiria S., Lechevallier Y., Gascuel O. & Canu S. (1997) *Statistique et Méthodes Neuronales*, Dunod.
- Van Cutsem (1994) *Classification and Dissimilarity Analysis*, Springer Verlag.
- Winograd T. (1983) *Language as a Cognitive Process*, vol. I: Syntax, Addison Wesley.
- Yang Y. (1998) An evaluation of statistical approaches to text categorization, *Information Retrieval*, p. 23-29.