*TALN 2004, Fès, 19-21 avril 2004*

# Catégorisation de patrons syntaxiques par Self Organizing Maps

Jean-Jacques Mariage et Gilles Bernard

Groupe CSAR, Laboratoire d'Intelligence Artificielle – Université Paris 8
2, rue de la Liberté, 93526 St Denis Cdx, France
jam@ai.univ-paris8.fr

## Résumé – Abstract

Dans cet article, nous présentons quelques résultats en catégorisation automatique de données du langage naturel sans recours à des connaissances préalables. Le système part d'une liste de formes grammaticales françaises et en construit un graphe qui représente les chaînes rencontrées dans un corpus de textes de taille raisonnable ; les liens sont pondérés à partir de données statistiques extraites du corpus. Pour chaque chaîne de formes grammaticales significative, un vecteur reflétant sa distribution est extrait et passé à un réseau de neurones de type carte topologique auto-organisatrice. Une fois le processus d'apprentissage terminé, la carte résultante est convertie en un graphe d'étiquettes générées automatiquement, utilisé dans un *tagger* ou un analyseur de bas niveau. L'algorithme est aisément adaptable à toute langue dans la mesure où il ne nécessite qu'une liste de marques grammaticales et un corpus important (plus il est gros, mieux c'est). Il présente en outre un intérêt supplémentaire qui est son caractère dynamique : il est extrêmement aisé de recalculer les données à mesure que le corpus augmente.

The present paper presents some results in automatic categorization of natural language data without previous knowledge. The system starts with a list of French grammatical items, builds them into a graph that represents the strings encountered in a reasonable corpus of texts; the links are weighted based upon statistical data extracted from the corpus. For each significant string of grammatical items a vector reflecting its distribution is extracted, and fed into a Self-Organizing Map neural network. Once the learning process is achieved, the resulting map will be converted into a graph of automatically generated tags, used in a tagger or a shallow parser. The algorithm may easily be adapted to any language, as it needs only the list of grammatical markers and a large corpus (the bigger the better). Another point of interest is its dynamic character: it is easy to recompute the data as the corpus grows.

## Keywords – Mots Clés

Langues naturelles, réseaux neuronaux, extraction de connaissances.
Natural languages, neural networks, knowledge extraction.

# 1 Introduction

Les résultats présentés ici sont la première étape d'un projet dont le but est d'étiqueter et d'analyser des textes volumineux en recourant le moins possible à des connaissances préalables qu'il est nécessaire de spécifier manuellement, particulièrement dans des contextes où des textes étiquetés à la main sont inexistants ou très rares.

Le seul moyen de prédire les étiquettes pour les mots inconnus est de se baser sur des mots connus et des règles grammaticales, les uns comme les autres étant spécifiés manuellement et reposant essentiellement sur des connaissances expertes ou des textes étiquetés au préalable, et donc sujets à contradiction, à la fois entre les types de documents et les domaines et entre experts.

Le but de notre système est, dans une première étape, d'automatiser la construction des règles grammaticales qui peuvent ensuite être affinées par l'expert. Les données initiales sont restreintes à des données grammaticales, ce qui constitue la partie essentielle de toute langue naturelle, et à un corpus le plus étendu possible. La langue considérée ici est le français, mais notre système est actuellement en cours d'application au grec et à l'arabe.

# 2 Collecte des données

La liste des items grammaticaux doit être réalisée manuellement, mais sa définition est relativement simple. Nous avons utilisé une liste de 311 items grammaticaux ; cette liste contient des items ambigus qui sont présents à la fois dans l'inventaire lexical et grammatical, comme *ton* en français, qui peut être un substantif ou un adjectif possessif (*ton* livre, son *ton*), et aussi des items grammaticaux dont la fonction est ambiguë (*le* livre et je *le* livre).

Dans le souci d'assurer la reproductibilité de nos expérimentations, nous avons sélectionné le corpus parmi les bases de documents français en libre accès, disponibles sur Internet[1]. Notre sélection comporte tous les textes écrits après 1750 (les textes récents sont rarement accessibles en raison des droits d'auteur). Ce corpus contient des textes électroniques de sources diverses (scannés ou saisis manuellement), comportant beaucoup d'erreurs. Il est composé de divers types de documents (surtout des romans, mais aussi des documents techniques et des périodiques). Nous avons réalisé les premières expérimentations sur une partie du corpus (environ 600 000 mots), mais les tests actuels sont exécutés sur un corpus dix fois plus important.

# 3 Construction du graphe de chaînes grammaticales

La première étape du processus, consiste à extraire les formes grammaticales du corpus et à en construire un graphe qui contient tous les liens possibles entre les items grammaticaux et leurs contextes.

---

[1] Ils proviennent essentiellement de l'Association des Bibliophiles Universels (ABU), dont le corpus est disponible en libre accès à l'adresse suivante : http://cedric.cnam.fr/ABU.

Chaque texte du corpus est découpé en paragraphes, les paragraphes en phrases, et les phrases en segments de phrases délimités par la ponctuation (chaque étape produisant un encodage SGML). L'étape de découpage en paragraphes a pour fonction essentielle de désambiguïser la ponctuation reliée à des débuts de type "A.1.2" et à l'usage des guillemets et des retours chariot. L'étape suivante de découpage en phrases désambiguïse la ponctuation finale ; la ponctuation qui subsiste est utilisée pour produire les segments de phrases.

Le processus entraîne une certaine quantité d'erreurs, due soit aux choix de programmation, soit, plus fréquemment, à des incohérences dans certains des fichiers d'entrée ; ainsi, certaines phrases comportent véritablement des paragraphes entiers, ou même plus. Mais cela ne semble pas affecter le résultat final.

Les mots des segments de phrases sont ensuite étiquetés comme suit : chaque mot qui ne figure pas dans la liste des items grammaticaux est associé à une étiquette parmi quatre possibles : (1) Mot initial de phrase en majuscules, (2) autre mot en majuscules, (3) nombre, et (4) mot en minuscules. L'étiquetage des items grammaticaux est réalisé dans leur forme en minuscules.

Les étiquettes sont extraites et les occurrences successives de la même étiquette sont remplacées par une seule étiquette. Le processus produit des patterns comme “ *1* *ne* *2* pas le *3* *4* ”, où *2*, par exemple, représente une séquence de mots de type 2. Les formes ne contenant que des étiquettes non grammaticales sont supprimées.

Approximativement 70.000 formes différentes sont générées ; les items grammaticaux ont environ 300.000 occurrences, et les 4 étiquettes lexicales mentionnés antérieurement ont de l'ordre de 200.000 occurrences (ces étiquettes représentent des *chaînes* d'items lexicaux, pas seulement des items lexicaux).

Un graphe est ensuite construit, avec un nœud pour chaque symbole rencontré dans le corpus, plus un nœud pour le symbole spécial représentant le début des segments de phrases, et un nœud pour la terminaison de segments de phrases.

Les liens en sortie de chaque nœud, excepté le nœud de terminaison, représentent l'arbre des symboles successifs : la racine est le symbole contenu dans le nœud et il y a autant de liens que de symboles rencontrés à la suite de ce symbole dans les patterns. Chaque lien est pondéré avec le nombre de fois où le symbole fils suit le symbole père dans les patterns.

(*a* occs
#
(*b* occs # (*d* occs #) (*e* occs #))
(*c* occs (*f* occs #)) )

où *a, b, c,* … sont les symboles, # est le symbole terminal, et occs, le nombre d’occurrences.

Chaque nœud reçoit autant de liens entrants qu'il y a d'arbres qui le contiennent. Les liens sortants sont étiquetés selon les liens entrants (ou le nœud racine). Pour chaque lien, un lien inverse est construit, de cette manière nous pourrons appliquer, dans une étape ultérieure, les programmes qui suivent non seulement aux successeurs de tout nœud donné, mais aussi à ses prédécesseurs.

# 4 Extraction des chaînes grammaticales

Les chaînes grammaticales sont extraites du graphe comme suit : nous sélectionnons chaque nœud qui contient un item grammatical. Pour chaque nœud, les étiquettes déterminent dans quelle mesure les liens entrants sont en relation avec les liens sortants. De cette manière, nous suivons les liens et extrayons les chaînes jusqu'au dernier nœud contenant un item grammatical ou jusqu'à ce que le dernier nœud significatif (voir ci-dessous) soit atteint. Les poids des liens sortants du dernier nœud de chaque chaîne, étiquetés avec le début de chaque chaîne, constituent un vecteur de la distribution des successeurs de la chaîne.

# 5 Calcul des vecteurs

Pour chaque lien sortant, la fréquence locale (le poids du lien divisé par la somme des poids de tous les liens sortants de la même étiquette) est calculée et divisée par la fréquence globale du nœud fils (le nombre total d'occurrences de son symbole, divisé par le nombre total d'occurrences de tous les symboles différents des symboles de début et de fin). Nous obtenons ainsi un vecteur de la déviation de la distribution locale par rapport à la distribution globale.

*a* : [dev(*a*) dev(*b*) dev(*c*) dev(*d*) dev(*e*) dev(*f*) dev(#)]

*ab* : [dev(*a*) dev(*b*) dev(*c*) dev(*d*) dev(*e*) dev(*f*) dev(#)]

*abd* : [dev(*a*) dev(*b*) dev(*c*) dev(*d*) dev(*e*) dev(*f*) dev(#)]

*ac* : [dev(*a*) dev(*b*) dev(*c*) dev(*d*) dev(*e*) dev(*f*) dev(#)]

où dev(*x*) = locfreq (*x*) / globfreq (*x*),

locfreq (*x*) = nombre d'occurrences de x dans le contexte (après *a*, *ab*, *abd*, *etc*.) / nombre d'occurrences du prédécesseur.

globfreq (x) = nombre d'occurrences de x / nombre d'occurrences de tous les symboles

Plutôt que de calculer simplement les fréquences locales, nous avons choisi de calculer la déviation, suite à des expérimentations antérieures (avec des techniques de regroupement par lien unique) qui montraient que les vecteurs de fréquences locales étaient trop fortement semblables pour être séparés dans l'étape suivante (un très petit nombre de neurones étaient sélectionnés pour la totalité de l'ensemble d'apprentissage). Cela est dû au caractère massif de la distribution des étiquettes génériques (représentant des données non grammaticales). L'élimination des composantes des vecteurs correspondant à des étiquettes lexicales a entraîné d'importantes pertes d'information (ainsi, la distribution de *Mr*, *Mlle* et des formes semblables reflète le fait qu'elles sont habituellement suivies par des mots en majuscules).

La méthode de calcul de la déviation que nous avons choisie présente toutefois un inconvénient : l'information apportée par la fréquence locale du symbole terminal (le nombre de fois où la fin de segment apparaît) est perdue, parce que parler de fréquence globale pour la fin de segment n'a aucune signification.

# 6 Sélection des vecteurs

Une déviation élevée peut être due à une fréquence locale importante ou à une faible fréquence globale, auquel cas les résultats peuvent être trompeurs. Le recours habituel à un seuil absolu ne semblait pas souhaitable, en ce qu'il éliminerait, par exemple, un mot qui apparaît 15 fois avec toujours le même successeur, ce qui semble être un résultat beaucoup plus significatif que 100 occurrences d'un mot polysémique avec 50 successeurs différents. Nous avons donc envisagé un seuil qui soit fonction de la polysémie de la chaîne de mots, telle qu'elle est indiquée par la longueur de son profil, *i.e*. le nombre de successeurs ayant une fréquence plus élevée qu'une valeur $\varepsilon$ donnée, choisie faible.

Le caractère significatif (l'importance statistique) Sign($s$) d'une chaîne est mesuré par la formule :

$$Sign(s) = \frac{(N(s) / M) - L(s)}{L(s)}$$

où $N(s)$ est le nombre d'occurrences de $s$, $L(s)$ la longueur de son profil, et $M$ la valeur minimale, considérée comme significative par successeur (15 dans l'exemple donné).

Les chaînes retenues sont celles dont le caractère significatif est supérieur à une valeur $\theta$ donnée, choisie petite.

Pour générer l'ensemble de vecteurs d'apprentissage à passer au réseau de neurones, nous avons fixé ces valeurs comme suit : $\varepsilon$ = 0.01, $M$ = 20, $\theta$ = 0. Pour produire les données utilisées en phase de test, destinées à évaluer la capacité de reconnaissance et de généralisation du réseau, nous avons adopté les réglages suivants : $\varepsilon$ = 0.05, $M$ = 10, $\theta$ = -0.1.

# 7 La carte topologique auto-organisatrice

Les vecteurs de données représentant les déviations des successeurs de chaque chaîne sont passés en entrée à un réseau de neurones de type carte topologique auto-organisatrice (noté ci-après SOM) de (Kohonen, 1982). SOM est un algorithme d'apprentissage non-supervisé qui cartographie les classes de données d'entrée en réalisant une projection, depuis leur espace multidimensionnel d'origine, dans l'espace interne de sa mémoire qui est bi-dimensionnel. L'algorithme construit un ordonnancement topologique des relations implicites qu'il découvre entre les classes de données. Sa représentation interne facilite l'analyse des relations entre classes, par la réduction dimensionnelle qu'elle leur applique, et minimise les effets de mauvaise classification. La structure bi-dimensionnelle de l'espace de représentation fournit une interface de visualisation conviviale où la similarité entre les classes de données est encodée dans la proximité entre les amas d'unités distribuée sur la carte : des formes reliées dans l'espace des données sont situées proches les unes des autres dans l'espace de la carte. Les principaux paramètres de ce modèle sont : le nombre de neurones, le maillage des unités de la carte, la topologie de la carte, tore ou plane, (Mariage, 1997), le taux d'apprentissage, le voisinage, et les fonctions qui déterminent sa décroissance dans l'espace et dans le temps, le rayon de propagation, le nombre d'itérations, le nombre de phases d'apprentissage.

Dans les expérimentations rapportées ci-dessous, les paramètres étaient réglés manuellement, avec deux phases d'entraînement. En première phase d'apprentissage (d'ordonnancement grossier), le nombre d'itérations était entre 25 % et 30 % du nombre d'itérations choisi en phase d'affinage. Nous avons utilisé une carte bi-dimensionnelle. Les unités étaient organisées en maillage hexagonal. L'entraînement était effectué avec prés de 7 000 vecteurs de 301 composantes présentés au réseau en ordre aléatoire. Les mémoires des unités étaient initialisées avec des valeurs aléatoires de ± 0.05 autour de la moyenne des valeurs des vecteurs de données. La règle d'activation était la distance euclidienne, la plus petite désignant l'unité de meilleur appariement. Le taux d'apprentissage $\alpha^{(t)}$, $0 < \alpha^{(t)} < 1$, évoluait en fonction du temps. Il était affecté d'une décroissance linéaire en $1 - (t / T_{total})$. En première phase d'entraînement, $\alpha$ était réglé à 0.5, tandis qu'en phase d'affinage, il avait une valeur de 0.05. Deux fonctions de voisinage ont été testées : le *bubble algorithm* (Kohonen, 1982, 1995) et le voisinage gaussien (Ritter, Martinetz et Schulten, 1989) sans présenter de différences significatives. Dans les deux cas, la taille du voisinage décroissait jusqu'à un rang autour de l'unité gagnante.

Plusieurs séries d'expérimentations ont été réalisées sur les mêmes données avec des configurations différentes de SOM. Les premiers essais avaient pour but d'estimer approximativement la catégorisation sur une carte réduite en fonction de la variation des paramètres de configuration de SOM. La carte comportait 96 unités (12 * 8). Le rayon de voisinage initial était choisi de manière à couvrir la totalité des unités en première phase. En seconde phase, un rayon de 5 rangs d'unités était adopté. L'entraînement était effectué pendant 36 000 itérations en première phase et 120 000 en phase d'affinage. Les résultats obtenus sont décrits dans la section 9 ci-dessous. Ils ont été établis manuellement en comptant les occurrences des chaînes grammaticales classées par les unités. Une deuxième série d'évaluations confirme ces résultats. La carte comportait 260 unités (26 * 10). Un voisinage initial de 7 rangs autour de l'élément actif était choisi en phase d'ordonnancement (≈ 87 % des unités). En phase d'affinage, un rayon de 3 rangs d'unités (≈ 19 %) était adopté. L'entraînement était effectué pendant 80 000 itérations en première phase et durant 320 000 en phase d'affinage. Nous donnons ci-dessous (section 8) les principales caractéristiques de la catégorisation obtenue. Une analyse automatique plus approfondie des résultats est en cours. Cette étape va nous permettre d'affiner encore les réglages des paramètres de configuration du réseau SOM et donc d'améliorer la résolution de la topologie.

# 8 Estimation de la qualité d'apprentissage

La qualité d'apprentissage était évaluée en fonction de deux critères : la résolution de la carte et la préservation de la topologie, calculés sur les deux ensembles de données d'apprentissage et de test. Dans les tableaux 1 et 2 ci-dessous, chaque ligne indique le meilleur résultat obtenu parmi 50 essais d'entraînement. Les nombres d'itérations sont exprimés en milliers.

L'erreur de quantification est la distance moyenne entre chaque vecteur de données et l'unité de meilleur appariement qu'il déclenche. Elle reflète la résolution de la carte topologique.

$$\sqrt{\sum_{x=0}^{X-1}\left(x_i - w_{bmu\,i}\right)^2} \Big/ X$$

L'erreur topographique est la proportion de vecteurs de données pour laquelle le premier et le second *bmus* ne sont pas adjacents. Elle rend compte de la préservation de la topologie.

$$\sum_{x=0}^{X-1} d_{node}\left(bmu_1 - bmu_2\right) \neq 1 \Big/ X$$

| Nombre d'itérations | Données d'entraînement | | Données de test | |
|---|---|---|---|---|
| | Erreur de quantification | Erreur topographique | Erreur de quantification | Erreur topographique |
| 10/40 | 71.675 | 0.0462 | 79.263 | 0.0464 |
| 20/80 | 67.665 | 0.0288 | 76.313 | 0.0368 |
| 30/120 | 65.502 | 0.0496 | 74.793 | 0.0570 |
| 40/160 | 63.936 | 0.0470 | 73.993 | 0.0459 |
| 50/200 | 63.556 | 0.0530 | 73.605 | 0.0544 |
| 60/240 | 62.842 | 0.0481 | 72.924 | 0.0626 |
| 70/280 | 61.805 | 0.0462 | 72.205 | 0.0509 |
| 80/320 | 61.558 | 0.0375 | 72.119 | 0.0429 |

Tableau 1. Estimation de la qualité d'apprentissage

La qualité d'apprentissage était mesurée à partir de la différence entre les deux meilleurs essais d'entraînement successifs, calculée comme : $(E^{(t-1)} - E^{(t)}) / E^{(t)}$, où $E$ est la mesure et $t$ l'indice temporel de l'essai. L'évolution de l'erreur était comparée à un seuil $S$ (un paramètre défini par l'utilisateur), utilisé comme critère d'arrêt. Ici, une valeur de 0.001 était choisie pour $S$. La qualité d'entraînement était estimée sur les données de test de manière à sélectionner les essais d'apprentissage ayant la meilleure capacité de généralisation.

| Nombre d'itérations | Données d'entraînement | | Données de test | |
|---|---|---|---|---|
| | Erreur de quantification | Erreur topographique | Erreur de quantification | Erreur topographique |
| 10/40 | - | - | - | - |
| 20/80 | 0.0559 | 0.604 | 0.0372 | 0.261 |
| 30/120 | 0.0320 | -0.419 | 0.0199 | -0.354 |
| 40/160 | 0.0239 | 0.055 | 0.0107 | 0.195 |
| 50/200 | 0.0059 | -0.113 | 0.0052 | -0.185 |
| 60/240 | 0.0113 | 0.102 | 0.0093 | -0.131 |
| 70/280 | 0.0168 | 0.041 | 0.0010 | 0.230 |
| 80/320 | 0.0040 | 0.232 | 0.0012 | 0.186 |

Tableau 2. Evolution des mesures d'erreur

Le seuil de diminution de l'erreur était atteint pour 70/280 mille itérations d'entraînement. Un essai supplémentaire a été réalisé avec 80/320 mille itérations. La stagnation de l'erreur de quantification confirme la bonne résolution de la carte sans l'améliorer significativement. L'erreur topographique indique le meilleur degré de préservation de la topologie obtenu.

# 9 Catégories obtenues

Dans plusieurs de nos essais, les résultats furent plutôt surprenants en ce qu'une grande quantité de chaînes grammaticales étaient rassemblées selon l'influence d'un élément dans la chaîne, parfois le dernier, mais dans d'autres cas le plus important quelle que soit sa position. Ainsi, des chaînes contenant *je*, marque ayant une forte capacité prédictive sur son contexte, étaient concentrées dans un amas de quatre à cinq unités (Cf. Figure 1 ci dessous). Les chaînes finissant par un syntagme nominal complet étaient classées dans un groupe d'unités couvrant environ les deux tiers de la carte, avec de part et d'autre, les chaînes incomplètes réparties dans deux zones distinctes de moindre importance. Un autre fait intéressant est que la distribution des chaînes grammaticales situées en début de segment de phrase reflète avec précision leur distribution globale : les chaînes contenant le symbole initial étaient généralement classées dans la même unité que les chaînes ne le comportant pas. La classification groupait ensemble les pronoms de la troisième personne du singulier et les chaînes qui les contenaient, mais dans des unités séparées. Ainsi *il*, *elle*, *on* étaient dans la même unité, et *qu'il*, *qu'elle*, *qu'on* étaient dans une autre unité (proche), groupés avec *qui*, et *puis il*, *puis elle* étaient encore dans une autre. Ainsi, étaient préservées à la fois la relation entre *il-elle-on*, mais aussi l'influence de certains des marqueurs grammaticaux environnants se faisait sentir et produisait une catégorie différente (même si elle est située à proximité). Dans un amas contigu, il en était de même pour les pronoms de la troisième personne du pluriel (*ils*, *elles*). Les chaînes terminées par des déterminants possessifs *son-sa-vos*, etc. étaient différenciées des autres déterminants. Les formes verbales étaient nettement séparées des formes nominales. A l'intersection de ces deux zones, huit unités encodaient les formes verbales réflexives. Une forte proportion des unités qui se déclenchaient préférentiellement pour des syntagmes nominaux, encodaient un nombre de formes nettement supérieur à la moyenne des autres unités, reflétant la plus forte densité des syntagmes nominaux dans la langue.


Figure 1 : Exemple de topologie des principales catégories obtenues.

Le fait que SOM capture effectivement ces différences, et d'autres de même nature, fut une grande surprise, et nous n'y avons pas encore trouvé une explication convaincante. Par

exemple, la différence entre les pronoms singuliers et pluriels ne pouvait pas être découverte facilement, car les pronoms de la troisième personne sont suivis par les mêmes marqueurs indépendamment de leur nombre (ex. *il le* / *ils le*, *il ne* / *ils ne*, etc.), et aucune information n'est donnée dans les items lexicaux : Les deux sortes sont suivies de l'étiquette générique *4*. Seule possibilité d'explication : l'influence de l'auxiliaire (portant les marques de nombre). Le phénomène est encore plus curieux pour les déterminants possessifs : nous ne voyons pas quel contexte peut différencier *son* et *le*. La même chose se produit pour les formes d'*être* et *avoir*. La seule hypothèse que nous pouvons proposer est que ces différences n'exercent pas d'influence sur les catégories de successeurs de ces chaînes, mais qu'elles influencent la distribution locale de ces catégories ; nous persistons à rechercher des explications à cette influence. D'autres résultats étaient plus aisément explicables : toutes les chaînes négatives du type "*il ne se *4**" sont rassemblées dans un amas, essentiellement en raison de la fréquence de la marque négative *pas* qui suit normalement ces chaînes ; les noms propres déterminants (*Mr* et semblables) étaient groupés ensembles, à cause de la fréquence des mots avec majuscule à l'initiale...

A la suite de ces premiers résultats plutôt encourageants, une entreprise de classification manuelle des chaînes d'entrées est en cours. Cette étape va nous permettre de sélectionner au mieux les paramètres de configuration du réseau SOM et d'évaluer plus finement la qualité des résultats.

# 10 Vers l'étiquetage

Quelle que soit l'explication de ces ordonnancements topologiques, une chose est sûre : les chaînes grammaticales peuvent être groupées et différenciées en considérant uniquement la distribution des formes qui les suivent. Après avoir considéré ici uniquement le contexte suivant, nous entendons maintenant explorer la classification des même chaînes en prenant en compte à la fois les prédécesseurs et les successeurs, tout en affinant notre chaîne de traitement actuelle ; et augmenter significativement la taille de notre corpus initial. Des résultats d'expérimentations antérieures sur la délimitation des constituants de phrase réalisée avec le processus stochastique récurrent de Harris (1968), peuvent aussi être intégrées ici. Les catégories de formes grammaticales produites par notre chaîne de traitement, avec les perfectionnements nécessaires mentionnés ci-dessus, peuvent être utilisées comme étiquettes de la manière suivante : les propriétés des chaînes élémentaires d'un mot sont bien connues (*il* est suivi par un verbe, *du* par un nom) et peuvent être propagées aux chaînes regroupées dans la même unité.Mais, peut-être plus important encore, les items ambigus cessent d'être ambigus dans les chaînes (ou au moins leur ambiguïté diminue de manière importante). Sans qu'il lui soit donné de règle, notre système a découvert implicitement ce phénomène, comme le montre le fait qu'il classe les chaînes contenant des items ambigus (en particulier *le*, *la*, *les*, comme article ou comme pronom), dans des groupes complètements différents, en séparant nettement les articles, réunis avec d'autres déterminants, et les pronoms (bien que le regroupement pour ces derniers soit moins clair, sans doute à cause de la faible quantité d'emploi de *le la les* en tant que pronom).

Revenons, en conclusion, sur la grande simplicité de notre principe de traitement : qui est la condition de sa portabilité (à d'autres langues) et de sa reproductibilité, tout comme la condition requise pour laisser la structure intrinsèque des données émerger.

## Références

BERNARD G. (1997), Experiments on distributional categorization of lexical items with Self Organizing Maps, *Proceedings of WSOM'97*, Helsinki, pp. 304-309.

BERNARD G. (2003), Détection automatique de structures syntaxiques, *Proceedings of the 8th International Symposium on Social Communication*, Santiago de Cuba.

BRILL E. (1997), Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging, *Natural Language Processing Using Very Large Corpora*, Kluwer Academic Press.

BRILL E. 1995, Unsupervised learning of disambiguation rules for part of speech tagging.

BRISCOE J. (1994), Prospects for practical parsing: robust statistical techniques, *Corpus-based Research into Language*: A Feschrift for Jan Aarts, de Haan & Oostdijk, Ed, Amsterdam.

HARRIS Z. (1968), *Mathematical structures of language*, John Wiley & Sons, New York.

JONES B. (1994), Can punctuation help parsing?, *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto, Japan.

JOSHI A. K. (1985), How much context-sensitivity is necessary for characterizing structural descriptions Tree adjoining grammars?, *Natural Language Processing Theoretical, Computational and Psychological Perspectives*, Dowty, Karttunen, Zwicky, Ed, Cambridge University Press, New York.

JOSHI A. K. (1987), The convergence of mildly context-sensitive grammatical formalisms, *Processing of Linguistic Structure*, Santa Cruz.

KOHONEN T. (1982), Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 43, 59-69.

KOHONEN T. (1995), *Self Organizing Maps*, Springer, Heidelberg.

LARI K., YOUNG S. J. (1990), The estimation of stochastic context-free grammars using the Inside-Outside algorithm, *Computer Speech and Language Processing*.

MARIAGE J.-J. (1997), Dynamic neighborhoods in Self Organizing Maps, Proceedings of *WSOM'97*, Helsinki, pp. 175-180.

MERIALDO B. (1994), Tagging English Text with a Probabilistic Model, *Computational Linguistics* (20), p. 155.

RITTER H. J., MARTINETZ T. M., SCHULTEN K. J. (1989), Topology conserving maps for learning visuo-motor coordination, *Neural Networks*, Vol. 2 (3), pp. 159-168.

SCHUETZE H. (1995), *Distributional Part-of-Speech Tagging*, in EACL 7.