La classification non supervisée (clustering) de documents textuels par les automates cellulaires

Hamou Reda Mohamed [1], Ahmed Lehireche [2]
Laboratory Evolutionary Engineering and Distributed Information Systems
University Djillali Liabes of Sidi Bel Abbes

[1] Université Dr MOULAY Tahar de Saïda, Faculté des sciences et technologie, Département d'Informatique, Tel : 0554380115, hamoureda@yahoo.fr

[2] Université Djillali Liabes de Sidi Bel Abbes, Faculté des science de l'ingénieur, Département d'Informatique, elhir@yahoo.com

Résumé : Dans cet article nous présentons un automate cellulaire (Class_AC) pour résoudre un problème de text mining en l'occurrence la classification non supervisée (Clustering). Avant de procéder à l'expérimentation par l'automate cellulaire, nous avons vectorisés nos données en procédant à l'indexation des documents textuels provenant de la base de donnée REUTERS 21578 par l'approche Wordnet. L'automate que nous proposons dans cet article est une grille de cellules de structure plane avec un voisinage découlant de cette structure (planaire). Trois fonctions de transitions ont servi à faire varier l'automate ayant quatre états pour chaque cellule. Les résultats obtenus montrent que la machine virtuelle à calcul parallèle (Class_AC) regroupe efficacement des documents similaires à un seuil prés.

Mots Clés : Classification des données, Automates cellulaires, Méthodes biomimétiques, Data mining, Clustering et segmentation, Classification non supervisée.

1 – Introduction

Le biomimétisme dans un sens littéraire est l'imitation de la vie. La biologie a toujours été une source d'inspiration pour les chercheurs dans différents domaines. Ces derniers ont trouvé un modèle presque idéal dans l'observation des phénomènes naturels et leur adaptation en vue de résoudre des problèmes. Parmi ces modèles on trouve les algorithmes génétiques, les colonies de fourmis, les essaims particulaires, nuages des insectes volants **[Mon 2003]** et bien entendu les automates cellulaires qui sont l'objet de notre étude. Les premières approches citées sont des méthodes reconnues et largement étudiées par contre les automates cellulaires sont des méthodes très peu utilisées et notamment dans le domaine de la classification non supervisée et ça a été notre motivation quant à l'utilisation de cette méthode dans ce domaine. Cette méthode est connue de la communauté scientifique comme étant un outil d'implémentation de machine et autre (Un **Automate Cellulaire** (AC) est avant tout une machine formelle) par contre dans cet article la méthode des automates cellulaires est utilisée comme étant une méthode biomimétique.
Depuis les années 50, le biomimétisme n'a cessé de progresser de façon constante et est un des principaux enjeux de la recherche actuelle.

Le biomimétisme est une pratique scientifique consistant à imiter, ou à s'inspirer de systèmes naturels, ou vivants. Parmi les exemples de ce domaine, on retrouve entre autres : formes de poissons pour l'aérodynamisme de voitures, ou autres véhicules, ou encore l'algorithme de colonies de fourmis pour la recherche du plus court chemin dans un graphe...
Le Text mining, est l'ensemble des techniques et des méthodes destinées au traitement automatique des données textuelles en langage naturel, est une analyse multidimensionnelle des données textuelles qui vise à analyser et découvrir des connaissances et des relations à partir des documents disponibles. Dans le text mining les similarités sont utilisées pour produire des représentations synthétiques de vaste collection de documents. Le text mining comprend une succession d'étapes permettant de passer des documents au texte, du texte au nombre, du nombre à l'analyse, de l'analyse à la prise de décision.
Nous commençons par un état de l'art, les techniques d'indexations des documents utilisées, une description générale de l'automate cellulaire pour le clustering, des résultats et nous donnerons une conclusion et les perspectives.

Etat de l'art

Pour mettre en œuvre des méthodes de classification il faut faire un choix d'un mode de représentation des documents **[Seb 2002]**, car il n'existe actuellement aucune méthode d'apprentissage capable de représenter directement des données non structurées (textes). Ensuite, il est nécessaire de choisir une mesure de similarité et enfin, de choisir un algorithme de classification non supervisée.

a. Représentation des documents textuels

Un document (texte) di est représenté par un vecteur numérique de la façon suivante :

$$di = (V_{1i}, V_{2i}, ..., V_{|T|i})$$

Où T est l'ensemble des termes (ou descripteurs) qui apparaissent au moins une fois dans le corpus.
(|T| est la taille du vocabulaire), et V_{ki} représente le poids (ou la fréquence).
La représentation la plus simple des documents textuels est appelé «représentation sac de mots » **[Aas 1999]**, elle consiste à transformer des textes en vecteurs où chaque élément représente un mot. Cette représentation de textes exclut toute forme d'analyse grammaticales et de toute notion de distance entre les mots.
Une autre représentation, appelée "sac de phrases", assure une sélection de phrases (séquences de mots dans les textes, et non pas le lexème "phrases"), en favorisant ceux qui sont susceptibles de porter une signification. Logiquement, une telle représentation doit fournir de meilleurs résultats que ceux obtenus par la représentation "sac de mots".
Une autre méthode pour la représentation des textes est celle basée sur les techniques de lemmatisation et consiste à chercher la racine lexicale d'un terme **[Sah 1999]**, par exemple, la forme de l'infinitif singulier pour les verbes et les noms.
Une autre méthode de représentation, qui a plusieurs avantages (principalement, cette méthode traite les documents textuels indépendamment de la langue utilisée), est basée sur les "n-grammes" (un "n-gramme" est une séquence de n caractères consécutifs).

Il existe différentes méthodes pour calculer le poids V_{ki} sachant que, pour chaque terme, il est possible de calculer non seulement sa fréquence dans le corpus, mais aussi le nombre de documents contenant ce terme.
La plupart des approches **[Seb 2002**] sont centrées sur la représentation vectorielle des textes en utilisant la mesure TF x IDF.
TF : représente « Term Frequency » : nombre d'occurrences du terme dans le corpus.
IDF : représente le nombre de documents contenant le terme. Ces deux concepts sont combinés (par produit), en vue d'attribuer un plus fort poids aux termes qui apparaissent souvent dans un document et rarement dans l'ensemble du corpus.

b. Mesure de similarité

Plusieurs mesures de similarité entre documents ont été proposées dans la littérature en particulier on trouve la distance euclidienne, Manhattan et Cosinus que l'on détaillera dans la section 3.

c. Algorithme de classifications non supervisée

La classification non supervisée ou "clustering" est l'une des techniques fondamentales de l'extraction de données structurées ou non structurées. Plusieurs méthodes ont été proposées:
Classification hiérarchique : arbre de classes
- Classification hiérarchique ascendante : Agglomérations successives
- Classification hiérarchique descendante : Divisions successives

Classification à plat : algorithme des k-moyennes : Partition

Quelques travaux dans le domaine de la classification

- ✓ Un survol des algorithmes biomimétiques pour la classification réalisé dans Laboratoire d'Informatique de l'Université de Tours, Ecole Polytechnique de l'Université de Tours **[AZZ 2004]**.
- ✓ Classification de données par automate cellulaire **[AZZ 2005]**.
- ✓ Fouille visuelle et classification de données par nuage d'insectes volants. **[Mon 2003]**.
- ✓ Compétition de colonies de fourmis pour l'apprentissage supervisée : CompetAnts. **[VER 2005]**.
- ✓ Classification non supervisée contextualisée **[CAN 2004]**.
- ✓ SOM pour la Classification Automatique Non supervisée de Documents Textuels basés sur Wordnet **[Amine et al., 2008]**.

Ce sont là des articles les plus en vus pour l'inspiration de notre travail et surtout celui de la classification de données par automate cellulaire car d'autres travaux non cités dans cet article on fait objet de recherches bibliographiques mais cités en références.

2- Les techniques d'indexation utilisées

Nous avons utilisés dans notre expérimentation le corpus REUTERS 21578 qui représente une base de données de dépêches d'information en langue anglaise. Ainsi pour faire du clustering des documents textuels on doit faire un certain traitement

pour vectoriser (numériser) nos textes (sans perdre la sémantique) et appliquer ensuite notre automate cellulaire. La première étape de l'indexation est le prétraitement qui consiste à éliminer tout symbole qui ne correspond pas à une lettre de l'alphabet (points, virgules, traits d'union, chiffres etc.). Cette opération est motivée par le fait que ces caractères ne sont pas liés au contenu des documents et ne change rien au sens s'ils sont omis et par conséquent ils peuvent être négligés. La deuxième étape est appelée stopping qui correspond à la suppression de tous les mots qui sont trop fréquents (ils n'aident donc pas à distinguer entre les documents) ou jouent un rôle purement fonctionnel dans la construction des phrases (articles, prépositions, etc.). Le résultat du stopping est que le nombre de mots dans la collection, ce qu'on appelle la masse des mots, est réduit en moyenne de 50%. Les mots à éliminer, connus comme *stopwords*, sont récoltés dans la *stoplist* qui contient en général entre 300 et 400 éléments puis vient l'étape du stemming qui consiste à remplacer chaque mot du document par sa racine comme par exemple : national, nationalité et nationalisation sont remplacés par leur racine « national » et les verbes conjugués par leur infinitifs. Le stemming n'a pas d'impact sur la masse des mots, mais réduit de 30% en moyenne la taille du document. Nous avons utilisés l'algorithme de PORTER pour remédier à cette étape. Ensuite nous avons la lemmatisation en utilisant l'approche Wordnet qui représente une base de donnée lexicale, un dictionnaire informatisé développée par des linguistes. Les mots dans WORDNET sont représentés par leur forme canonique ou encore appelé lemme. Cette étape est utilisée pour préparer la suivante qui est l'étape cruciale de l'indexation à savoir la vectorisation (numérisation). La lemmatisation consiste à remplacer chaque mot du document par son synset (synonyme dans la base lexicale). Nous avons utilisés WORDNET comme base de donnée lexicale (car REUTERS 21578 est un corpus de dépêches en anglais). La vectorisation est réalisé par la méthode TF-IDF (Term Frequency / Inverse Document Frequency) qui est dérivé d'un algorithme de recherche d'information. L'idée de base est de représenter les documents par des vecteurs et de mesurer la proximité entre documents par l'angle entre les vecteurs, cet angle étant donc supposé représenter une distance sémantique. Le principe est de coder chaque élément du sac de mot par un scalaire (nombre) appelé tfidf pour donner un aspect mathématique aux documents textes.

$$tfidf = tf(i, j).idf(i) = tf(i, j).\log\left(\frac{N}{N_i}\right)$$

où :

- tf(i,j) est le term Frequency : fréquence du terme ti dans le document dj
- idf(i) est l'inverse document frequency : le logarithme du rapport entre le nombre N de documents dans le corpus et le nombre N_i de documents qui contiennent le terme ti .

Un document d_i du corpus après vectorisation est :
$d_i = (x_1, x_2, \ldots\ldots, x_m)$ où m est le nombre de mot du $i^{éme}$ sac de mot et x_j est son tf-idf
Ce schéma d'indexation donne plus de poids aux termes qui apparaissent avec une haute fréquence dans peu de documents. L'idée sous-jacente est que de tels mots aident à discriminer entre textes ayant différent sujet. Le tf-i*df* a deux limites fondamentales : La première est que les documents plus longs ont typiquement des poids plut forts parce qu'ils contiennent plus de mots, donc « les term frequencies » tendent à être plus élevées. La deuxième est que la dépendance de la « term

frequency » est trop importante. Si un mot apparaît deux fois dans un document d_j, cela ne veut pas nécessairement dire qu'il a deux fois plus d'importance que dans un document d_k où il n'apparaît qu'une seule fois.

3- L'automate cellulaire pour le clustering

L'automate que nous proposons est un réseau de cellules dans un espace à 2D et appartient à la famille (k,r) où k est le nombre d'états possibles d'une cellule c'est-à-dire le cardinal de l'ensemble d'états et r est l'environnement de la cellule i-e r représente le rayon de voisinage.

Cet automate possède 4 états possibles (k=4) et le rayon de voisinage est une seule cellule (r=1) i-e le voisinage utilisé n'est rien d'autre que le voisinage de Moore (8 cellules voisines autour de la cellule elle-même) légèrement modifié.

Donc une cellule de l'automate est morte, vivante, isolée ou contient une donnée d'où l'ensemble des états de l'automate est {Morte, Vivante, Isolée, Active}.

Une cellule morte contiendra la valeur 0, une cellule vivante contiendra la valeur -1, une cellule isolée contiendra la valeur -2 et une cellule active contiendra une donnée (numéro ou index du document du corpus).

Nous avons utilisés ces valeurs et surtout la valeur de la cellule vivante (-1) pour ne pas confondre entre une cellule vivante contenant la valeur 1 et une cellule contenant une donnée (numéro ou index du document) 1. Donc une cellule contiendra une valeur de l'ensemble {-2, -1, 0, 1, 2, ….., N} où N est le numéro du dernier document du corpus utilisé.

Si N est le nombre de documents à classifier, la taille de la grille cellulaire 2D est m x m sachant que m = 2 x (Int (Racine(n))+1) avec Int représentant la partie entière d'un nombre réel, Racine représentant la racine carrée et 2 représente un coefficient empirique pour avoir de la place pour l'organisation spatiale des classe dans la grille.

Exemple : Si on veut classifier 50 documents du corpus REUTER 21578 on doit avoir une grille de 8 x 8 pour représenter les 50 documents et une grille 16 x 16 pour représenter les différentes classes des 50 documents espacées dans la grille (Fig 3.a).

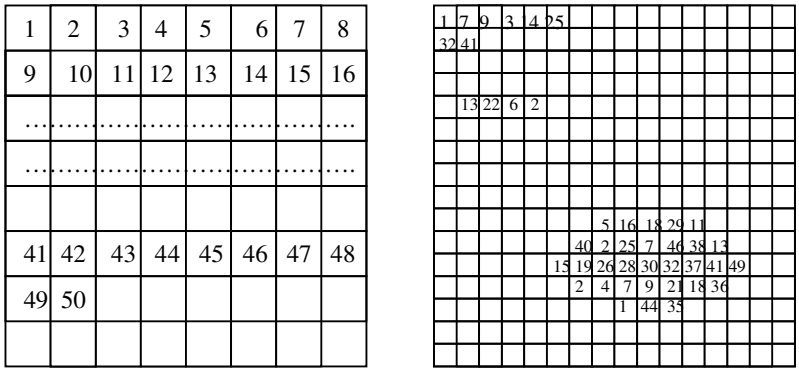


Fig 3.a : Exemple de grille pour 50 documents

3-1 Le Voisinage

Le voisinage utilisé dans l'automate que nous proposons est un voisinage hybride contenant le voisinage de Moore qui est le voisinage de rayon 1 contenant 8 cellules autour de la cellule elle-même et deux voisinages de rayon de 1 découlant du fait que la grille est planaire. Puisque la grille est planaire le voisinage des quatre extrémités contient seulement trois (3) cellules voisines et le voisinage d'une cellule (i,j) appartenant au périmètre de la grille (Sans les extrémités) est l'ensemble de cinq (5) cellules avoisinant la cellule (i,j) de rayon 1. (Fig 3.b)

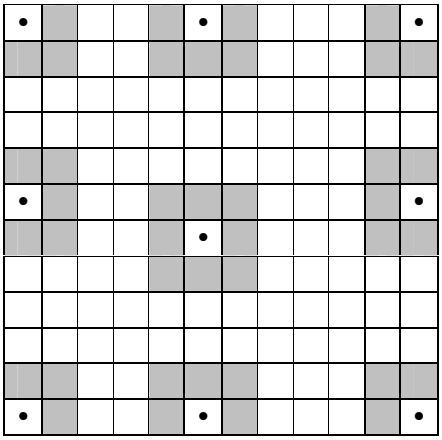


Fig 3.b : Voisinage

3-2 La fonction de transition de l'automate Class_AC

Règle 1 :

Si la cellule $C_{i,j}$ est morte alors la cellule $C_{i,j}$ ⟵ donnée
Voisinage $C_{i,j}$ devient vivant

Règle 2 :

Si cellule $C_{i,j}$ Vivante alors Vérifier voisinage
| Si voisinage contient au moins une cellule active
| Alors
| | $C_{i,j}$ ⟵ donnée similaire
| | Voisinage $C_{i,j}$ devient vivant
| Sinon
| | Voisinage $C_{i,j}$ devient isolé
| Fin
Fin

Règle 3 :

Si une cellule est isolée alors inchangé (Reste isolée)

3-3 La matrice de similarité

Nous avons expérimentés notre classification en utilisant trois (3) distances différentes en l'occurrence la distance euclidienne, Manhattan et cosinus.

3-3-1 : La distance Euclidienne
Distances entre vecteurs T_i et T_j dans espace multidimensionnel est

$$D(T_i, T_j) = \sqrt{\sum_k (x_k(T_i) - x_k(T_j))^2}$$

3-3-2 : La distance Manhattan
Distances entre vecteurs T_i et T_j dans espace multidimensionnel est

$$D(T_i, T_j) = \sum_k | (x_k(T_i) - x_k(T_j)) |$$

3-3-3 : La distance Cosinus
Distances entre vecteurs T_i et T_j dans espace multidimensionnel est

$$Cos(T_i, T_j) = \frac{T_i . T_j}{|| T_i || . || T_j ||}$$

Où T_i . T_j représente le produit scalaire des vecteurs T_i et T_j
$\| T_i \|$ et $\| T_j \|$ représentent respectivement les normes de T_i et T_j
La matrice de similarité est une matrice symétrique de dimension N x N, où N est le nombre de documents à classifier, de diagonale nulle (pour les distances euclidiennes et Manhattan) et de diagonale égale à 1 (pour la distance cosinus), et dont les indices représentent les numéro (index) des documents du corpus à classifier.

3-4 Description de l'algorithme Class_AC

- Indexer les documents du corpus à classifier.
- Vectoriser chaque document texte du corpus par la méthode TF-IDF.
- Calculer la matrice de similarité à partir des vecteurs trouvés : sim(i,j)=D(d_i,d_j).
- Initialiser toutes les cellules de l'automate à l'état « Morte » (état=0).
- **Répéter** (à chaque instant t)
- **Pour** chaque cellule de l'automate faire
  Si cellule est morte **Alors**

Cellule devient Active
Voisinage cellule devient Vivante

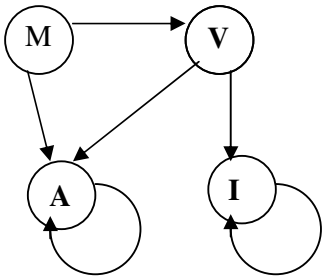
Fin Si

Si cellule est vivante **Alors**
Vérifier voisinage
Si voisinage contient au moins 1 cellule active
Alors
Cellule devient active (Donnée Similaire)
Voisinage cellule devient vivant
Sinon
Voisinage cellule devient isolé

Fin Si
Si Cellule est isolée **Alors** Cellule reste isolée (Inchangé)

- Fin Pour
- **Jusqu'à** Fin donnée.

A chaque itération de l'algorithme, les cellules vont changés leurs état selon les règles de transition définies par l'automate cellulaire qui vont tendre à regrouper des états similaires pour les cellules actives (contenant l'index des documents).La classification est recouvrante (Les données peuvent apparaître plusieurs fois dans la grille).



M : état de cellule morte

V : état de cellule vivante

A : état de cellule active

I : état de cellule isolée

Fig 3-4-a : Schématisation de l'automate

4- Expérimentation

Après expérimentation de l'algorithme Class_AC, sur des documents issus du corpus Reuters 21578, nous avons obtenu les résultats suivants en nombre de classes et pureté des clusters.
En ce qui concerne la pureté d'un cluster nous avons utilisés un seuil de similarité qui représente la distance entre deux documents. Si cette distance est inférieure ou égale au seuil alors les documents sont similaires. Pour la distance cosinus ce seuil est comparé à la valeur $|1 - \cos(V_i,V_j)|$.

Puisqu'on a utilisé un seuil donc on n'aura pas besoin de calculer l'entropie qui mesure la pureté du cluster trouvé ni de la F-mesure pour évaluer les taux d'erreurs en classification.

Définition du seuil

Seuil 1 : Pour les distances euclidiennes et Manhattan et après normalisation de la matrice de similarité (distance comprise dans [0,1]) nous avons tolérés un taux d'erreur de 10% (seuil 1=0,1) et pour la distance cosinus nous avons tolérés 20%.
Seuil 2 : Un seuil de 15% (seuil 2=0,15) pour les distances euclidiennes et Manhattan par contre seuil 2=0,25 (25%) pour la distance cosinus.
Ces valeurs de seuil ont été choisies après expérimentation de la classification par l'automate cellulaire.

Commentaires

Nous avons expérimenté notre automate cellulaire sur le corpus REUTERS 21578, nous avons procédé à l'extraction des 50 premiers textes que nous avons indexés. On a ensuite calculé leur matrice de similarité.

En terme de résultats (Tab1 et Tab2), nous avons obtenus différentes classes par les trois distances utilisées en variant le seuil de similarité. Les classes trouvées correspondent à un regroupement de documents similaires guidé en quelque sorte par le seuil établi (Tableau : Tab1). En terme de pureté du cluster, la similarité intra classe n'est rien d'autre que le seuil car la distance entre deux documents d'une même classe doit être inférieure ou égale au seuil, et la distance entre deux documents de classe différente est supérieure strictement au seuil (la similarité extra classe). Donc on n'avait pas à résoudre un problème de recherche opérationnelle sous contrainte (minimiser la similarité intra classe et maximiser la similarité extra classe) mais simplement choisir un bon seuil pour avoir une bonne classification.

Tab1 : Résultats de classification (Cosinus, Seuil 2)

Automate Cellulaire (Distance Cosinus-Seuil 2)	
Classe	**Documents**
0	1
1	2,16,17,23,26,37
2	3,9,13,23,26,37
3	4,16,17
4	5,8,25,39
5	6,18,42
6	7,10,12,22
7	11
8	14
9	15
10	19
11	20,26,37
12	21

Automate Cellulaire (Distance Cosinus-Seuil 2)	
Classe	**Documents**
13	36,40,42
14	24,25,39
15	27
16	28
17	29
18	30,39
19	31
20	32,33,43
21	34
22	35,41,45
23	38
24	44

Tab 2 : Résultats de classification pour 50 documents

N = 50 Documents

Seuil 1

	Cosinus	Euclidienne	Manhattan
Nombre de Classes	40	39	5
Temps S apprentis sage	0,04688	0,0625	0,0625

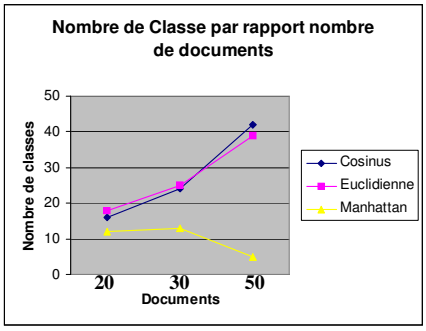
Seuil 2

	Cosinus	Euclidienne	Manhattan
Nombre de Classes	25	3	1
Temps S apprentis sage	0,04688	0 ,0625	0,0625

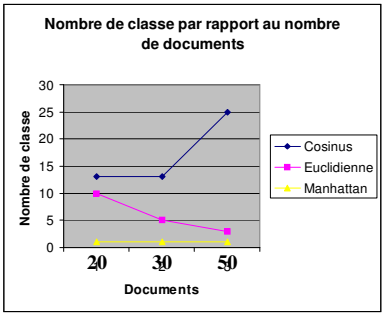
En terme de temps, la convergence de l'algorithme est très rapide (moins de 1 seconde) et par conséquent ce qui a été dit dans la littérature sur les automates cellulaires est respecté dans notre étude. Nous avons remarqués que le temps d'exécution était en croissance avec le nombre de documents. A titre indicatif l'expérimentation a été réalisée su PC Pentium IV cadencé 1,6 Mhz avec 512 Mo de mémoire vive.

Résultats de classification pour 20, 30 et 50 documents

Seuil 1



Seuil 2



5- Conclusion et perspectives

En conclusion, nous avons proposé un premier algorithme de classification non supervisée (Clustering) en utilisant les automates cellulaires. Après expérimentation nous avons prouvé que cet algorithme peut résoudre un problème de text mining qu'est le clustering en regroupant efficacement des documents textuels issus du corpus REUTERS. La fonction de transition utilisée dans notre automate le fait évolué en formant des groupes (cluster) similaires à un certain seuil prés. Les méthodes d'indexation des documents textuels tel que TF-IDF et l'approche Wordnet

nous ont aidés à numériser nos documents et ainsi pouvoir utiliser notre automate cellulaire sur des vecteurs numériques. Donc les passages des documents au texte, du texte au nombre, du nombre à l'analyse par les automates cellulaires et de l'analyse à la prise de décision sur la classification ainsi trouvée ont fait l'objet de cette étude dans cet article. Cet algorithme sera, dans le futur proche, comparé à un algorithme utilisant l'apprentissage par les cartes auto organisatrice de KOHONEN. L'algorithme peut contribuer ainsi à la problématique de la fouille de donnée textuelle et de la classification non supervisée.

Dans cet article, nous avons proposé un algorithme qui résout un problème de data mining en l'occurrence le text mining par une méthode biomimétique (Automates cellulaires). Cet algorithme sera dans le futur expérimenté pour d'autres types de données tels que les images et les données multimédias en général pour résoudre une autre problématique de fouille de données.

Références

[NEU 1966] VON NEUMANN J., *Theory of Self Reproducing Automata.*, University of Illinois Press, Urbana Champaign, Illinois, 1966.

[LUM 94] LUMER E., FAIETA B., *Diversity and adaption in populations of clustering ants.* In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3, pages 501-508. MIT Press, Cambridge, MA, 1994.

[BOC 1994] Efficient and effective clustering methods for spatial data mining. In J. BOCCA, M. JARKE & C. ZANIOLO, Eds., *20th Int. Conf. on Very Large Data Bases*, p. 144–155, Santiago, Chile : Morgan Kaufmann Publishers.

[APT 1994] APTÉ C., DAMERAU F., , WEISS S., Automated learning decision rules for text categorization, ACM Transactions on Information Systems, vol. 12, no 3, 1994, pp. 233-251.

[BUR 1998] BURGES C., A tutorial on Support Vector Machines for pattern recognition, Data Mining and Knowledge Discovery, vol. 2, no 2, 1998, pp. 121-1.

[Aas 1999] Aas, K., Eikvil, L.: Text categorization: a survey. *Technical report, Norwegian Computing Center*,1999.

[Sah 1999] Sahami, M.: Using Machine Learning to Improve Information Access. PhD thesis, Computer Science Department, Stanford University, 1999.

[Han 2000] J. Hansohm. Two-mode clustering with genetic algorithms. In Classification, Automation, and New Media: Proceedings of the 24th Annual Conference of the Gesellschaft Fur Klassifikation E.V., pages 87–94, 2000.

[Seb 2002] Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1),1–47, 2002.

[GAN 2003] GANGULY N., SIKDAR B. K., DEUTSCH A., CANRIGHT G., CHAUDHURI P. P., *A Survey on Cellular Automata*. Technical Report Centre for High Performance Computing, Dresden University of Technology, December 2003.

[Mon 2003] Nicolas Monmarché, Christiane Guinot,Gilles Venturini, Fouille visuelle et classification de données par nuage d'insectes volants, *Laboratoire d'Informatique de l'Université de Tours, École Polytechnique de l'Université de Tours - Département Informatique.*

[CAN 2004] Laurent Candillier, Isabelle Tellier, Fabien Torre. Tuareg : Classification non supervisée contextualisée - Université Charles de Gaulle - Lille 3 France.

[AZZ 2004] AZZAG H., PICAROUGNE F., GUINOT C., VENTURINI G., *Un survol des algorithmes biomimétiques pour la classification*. Classification Et Fouille de Donnée, pages 13-24, RNTI-C-1, Cépaduès. 2004.

[AZZ 2005] Classification de données par automate cellulaire, H. Azzag, F. Picarougne, C. Guinot, G. Venturini, *Université François-Rabelais de Tours, Laboratoire d'Informatique (EA 2101)*

[Aga 2005] Agata Kramm, AUTOMATES CELLULAIRES, Mémoire de maîtrise d'informatique, Universit´e Paris VIII, Septembre 2005

[AZZ 2005-A] H. Azzag, F. Picarougne, C. Guinot, G. Venturini. VRMiner: a tool for multimedia databases mining with virtual reality. Processing and Managing Complex Data for Decision Support (2005). J. Darmont and O. Boussaid, Editors.

[ALE 2006] Alessandro Vinciarelli, Indexation de Documents Manuscrits Offline

[Amine et al., 2008] A. Amine, Z. Elberrichi, M. Simonet, L. Bellatreche and M. Malki. SOM pour la Classification Automatique Non supervisée de Documents Textuels basés sur Wordnet. *Extraction et gestion des connaissances (EGC'2008)* – INRIA-Sophia Antipolis -France-. Volume 1. Revue des Nouvelles Technologies de l'Information RNTI-E-11 Cépaduès-Éditions 2008. ISSN : 1764-1667.