# Ordre et Désordre dans la Catégorisation de Textes

Simon Jaillet, Maguelonne Teisseire, Anne Laurent, Jacques Chauché

# Ordre et désordre dans la catégorisation de textes

**S. Jaillet, M. Teisseire, A. Laurent, J. Chauché**

*LIRMM UMR CNRS 5506 - Université Montpellier II*
*161, Rue Ada 34392 Montpellier Cedex 5, France*
*E-mail : {jaillet,laurent,teisseire,chauche}@lirmm.fr*

RÉSUMÉ. *La classification automatique de textes est une tâche adressée notamment par des approches statistiques à base de réseaux de neurones et de machines à vecteurs de support. Si ces approches permettent de réaliser de bons classifieurs au sens des mesures de classification, elles ne permettent pas de bénéficier de règles décrivant les décisions de classification. Or ces descriptions sont primordiales pour les experts démunis face aux grandes quantités de textes à analyser et traiter. Dans ce contexte, une approche à base de règles d'association a déjà été proposée par Bing Liu. Nous proposons dans cet article d'étendre cette approche par l'utilisation des motifs séquentiels avec la méthode SPaC (Sequential Patterns for Classification). La prise en compte de l'ordre des mots permet de représenter la succession de l'apparition des termes ou des concepts dans les textes Des expérimentations, menées sur des ensembles de textes en français et anglais, montrent l'intérêt de la méthode proposée. La prise en compte de l'ordre des mots par les motifs séquentiels mène toujours à de meilleurs résultats que les méthodes basées sur les règles d'association.*

ABSTRACT. *Text categorization is a well-known task essentially based on statistical approaches using neural networks, Support Vector Machines and other machine learning algorithms. Texts are generally considered as bags of words without any order. Although these approaches have proven to be efficient, they do not provide users with comprehensive and reusable rules about their data. These rules are however very important for users in order to describe the trends from the data they have to analyze. In this framework, an association-rule based approach has been proposed by Bing Liu (CBA). In this paper, we propose to extend this approach by using sequential patterns in the SPaC method (Sequential Patterns for Classification). Taking order into account allows us to represent the succession of words through a document without complex and time-consuming representations and treatments such as those performed in natural language and grammatical methods. We show on experiments that our proposition is relevant, and that it is very interesting compared to other methods.*

MOTS-CLÉS : *fouille de textes, catégorisation, motifs séquentiels*

KEYWORDS: *Text Mining, Categorization, Sequential Patterns*

## 1. Introduction

Les travaux sur la classification automatique de textes datent des années 60 [MAR 61]. Néanmoins, l'explosion du nombre de documents électroniques disponibles a mis en évidence le besoin de méthodes efficaces permettant de traiter de gros volumes de données. C'est pourquoi de nombreux travaux de recherche traitent cette problématique [SEB 02b, YAN 99, JéR 03]. Les résultats obtenus sont utiles aussi bien pour la recherche d'information que l'extraction de connaissances. L'objectif est de classer de façon automatique les documents dans des catégories préalablement définies soit par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore clustering [SAL 83, IWA 95].

Dans cet article, nous nous intéresserons plus particulièrement à la catégorisation de documents c'est-à-dire aux approches de classifications supervisées. La plupart des approches de catégorisation existantes reposent sur des méthodes considérées comme des approches statistiques [SEB 02a] telles que les Support Vector Machines (SVM) [VAP 95, JOA 98b]. Toutes ces approches se basent sur une mesure de la fréquence des mots telle que $TF$-$IDF$ (Term Frequency, Inverse Document Frequency). Néanmoins, même si les performances en terme de classification sont intéressantes, aucune de ces méthodes ne fournit un résultat compréhensible de la connaissance extraite et surtout ré-utilisable. Pour résoudre ce problème, une approche de classification basée sur les règles d'association, issues du domaine de la fouille de grandes bases de données, a été intialement proposée par Bing Liu (CBA) [LIU 98]. Les algorithmes de recherche de règles d'association s'avèrent très efficaces dans le contexte de gros volume de données, leur adaptation à la problématique de la catégorisation était donc très attractive. Cependant les performances comme classifieur n'étaient pas très satisfaisantes. Cette approche a donc été ensuite étendue et améliorée par [WAN 00], [LI 01], [JAN 03] et [BAR 03].

Toutes ces méthodes considèrent les textes comme des *sacs de mots* où aucun ordre n'est pris en compte lors du processus de classification. Aussi, il est intéressant de proposer une méthode prenant en compte une relation d'ordre entre les mots afin d'améliorer les résultats tout en conservant de bonnes performances pour de gros volumes de textes.

Nous proposons donc d'étendre les approches basées sur les règles d'association en prenant en compte une relation d'ordre entre les mots. Nous montrons que cet ordre permet d'améliorer le processus de classification sans le rendre plus complexe contrairement à d'autres extensions.

L'originalité de notre proposition, nommée SPaC (Sequential PAtterns for Categorization), est de rechercher non pas des règles d'association, mais des motifs séquentiels au sein des documents qui serviront ensuite de base à la classification. Introduits dans [AGR 95], les motifs séquentiels peuvent être vus comme une extension de la notion de règle d'association en y incluant une relation temporelle. Deux grandes étapes sont donc à considérer :

1) L'extraction des motifs séquentiels à partir de la base de documents. La granularité étant celle de la phrase c'est-à-dire que chaque document est considéré comme une suite ordonnée de phrases, elle-même constituée d'un ensemble de mots non ordonnés.

2) La construction d'un classifieur basé sur les motifs séquentiels préalablement obtenus.

Nous montrons qu'un classifieur basé sur les motifs séquentiels est très efficace, en particulier lorsque les classifieurs classiques comme les SVM sont en difficulté. De plus, un tel système possède trois qualités essentielles : (1) il s'appuie sur des règles compréhensibles et interprétables pour les utilisateurs finaux (contrairement à la grande majorité des systèmes de catégorisation comme les SVM, Rocchio, Naïve Bayes,...), (2) il permet de réaliser des analyses de tendance, comme proposées dans [LEN 97], suite aux différentes évaluations constatées dans les catégories et enfin (3) les motifs séquentiels sont plus précis et informatifs que les règles d'association. De plus, nous évaluons le classifieur obtenu non pas en terme de taux de réussite qui valorise le silence ou la non classification mais de la $F$-mesure [RIJ 79], la fonction la plus utilisée pour comparer les classifieurs [SEB 02b].

Cet article est organisé de la façon suivante. La section 2 définit les problématiques de la recherche de motifs séquentiels et de la catégorisation de textes. La section 3 présente un panorama des travaux basés sur l'utilisation de motifs fréquents pour la classification et l'utilisation des motifs séquentiels sur les textes. Notre méthode, SPaC, est détaillée section 4 puis les différentes expérimentations réalisées sur des jeux de documents en anglais et en français sont présentées à la section 5. Section 6, nous concluons et proposons quelques perspectives d'amélioration de SPaC.

## 2. Problématique

Nous allons tout d'abord définir le problème de la recherche de motifs séquentiels introduit dans [AGR 95] et étendu dans [SRI 96]. Ensuite, nous examinerons le problème de la catégorisation de texte.

### 2.1. *La recherche de motifs séquentiels*

Considérons $DB$ une base regroupant l'ensemble des achats réalisés par des clients. Chaque n-uplet $T$ correspond à une transaction financière et consiste en un triplet *(id-client, id-date, itemset)* : l'identifiant du client, la date de l'achat ainsi que l'ensemble des produits (items) achetés .
Soit $I = \{i_1, i_2, ..., i_m\}$ l'ensemble des $items$ (produits). Un *itemset* est un ensemble d'items non vide noté $(i_1 i_2 \dots i_k)$ où $i_j$ est un *item* (il s'agit d'une représentation non ordonnée). Une *séquence* $s$ est une liste ordonnée, non vide, d'itemsets notée $< s_1 s_2 ... s_p >$ où $s_j$ est un itemset. Une *n-séquence* est une séquence composée de $n$ items. Par exemple, considérons les achats des produits $1, 2, 3, 4, 5$, réalisés par le

client Dupont selon la séquence $s =< (1, 5) (2, 3) (4) (5) >$ indiquée table 1. Ceci signifie qu'hormis les achats des produits 1 et 5 puis 2 et 3 qui ont été réalisés ensemble, i.e. lors de la même transaction, les autres items de la séquence ont été achetés séparément. Dans notre exemple, $s$ est une 6-séquence.
Une séquence $< s_1 s_2 ... s_p >$ est une sous-séquence d'une autre séquence $< s'_1 s'_2 ... s'_m >$ s'il existe des entiers $i_1 < i_2 < ... i_j ... < i_n$ tels que $s_1 \subseteq s'_{i_1}, s_2 \subseteq s'_{i_2}, ..., s_p \subseteq s'_{i_n}$. Par exemple, la séquence $s' = <$ (2) (5) $>$ est une sous-séquence de $s$ car (2) $\subseteq$ (2, 3) et (5) $\subseteq$ (5). Toutefois, $<$ (2) (3) $>$ n'est pas une sous-séquence de $s$.

Tous les achats d'un même client sont regroupés et triés par date. Ils constituent la séquence de données du client. Un client *supporte* une séquence $s$ si $s$ est incluse dans la séquence de données de ce client ($s$ est une sous-séquence de la séquence de données). Le *support* d'une séquence $s$ est alors calculé comme étant le pourcentage des clients qui supportent $s$. Soit $minSupp$ le support minimum fixé par l'utilisateur, une séquence qui vérifie le support minimum (i.e. dont le support est supérieur à $minSupp$) est une *séquence fréquente*.

| Client | Date | Items |
|---|---|---|
| Dupont | 04/01/12 | Chocolat (5), TV (1) |
| Martin | 04/02/28 | Chocolat(5) |
| Dupont | 04/03/02 | Lecteur DVD (2) , Camescope (3) |
| Dupont | 04/03/12 | Imprimante (4) |
| Dupont | 04/04/26 | Chocolat (5) |

**Tableau 1.** *Exemple d'une base d'achats*

Le problème de la recherche de motifs séquentiels dans une base de données consiste à trouver les séquences maximales dont le support est supérieur au support minimum spécifié. Chacune de ces séquences est un motif séquentiel ou plus communément une séquence fréquente.

### 2.2. *La représentation des textes et la catégorisation*

La catégorisation de documents consiste à assigner une valeur booléeenne à chaque couple (document, catégorie). Pour cela, un processus de classification doit définir (1) une formalisation des textes et des catégories (2) une mesure entre les textes et les catégories (3) une politique de catégorisation permettant de décider si le texte est affecté ou non à la catégorie. La base de textes est partionnée en deux bases $T_{Train}$ et $T_{Test}$ où $T_{Train}$ constitue le jeu d'apprentissage à partir duquel le classifieur sera construit et $T_{Test}$ le jeu de test sur lequel le classifieur sera évalué.

La plupart des méthodes représentent les textes comme des sacs de mot [SEB 02b]. L'ordre n'est donc pas pris en compte. Un document est ensuite représenté par un vec-

teur dont chaque composant est un mot pondéré par une valeur numérique. La pondération la plus utilisée est $TF$-$IDF$ (Term Frequency - Inverse Document Frequency) définie par :

$$\text{Soit } w \text{ un mot, } \textit{tfidf}(w) = tf(w).log\frac{N}{\textit{df}\,(w)}$$

avec *tf*$(w)$ le nombre d'occurences de $w$ dans le document, *df*$(w)$ le nombre de documents contenant $w$ et $N$ le nombre total de documents. Le poids *tfidf*$(w)$ indique l'importance relative du mot dans le document. Des techniques de classification (SVM, naïve Bayes, $k$ plus proches voisins etc.) sont ensuite appliquées sur les vecteurs représentant les documents afin de déterminer la ou les catégories d'affectation.

## 3. Travaux associés

La fouille de textes a été très étudiée [LEN 97, ALI 97, AGR 00, SEB 02b]. Notre objectif dans cette section est de se focaliser sur la classification de textes et les motifs fréquents.

### 3.1. *Classification basée sur les règles d'association : la méthode CBA*

Dans [LIU 98], la méthode CBA basée sur les règles d'association est proposée. CBA est composée de deux modules, un générateur de règles (CBA-RG) basé sur l'algorithme Apriori [AGR 94] et un constructeur de classifieur basé sur les règles précédemment obtenues (CBA-CB).

#### 3.1.1. *Le générateur de règles CBA-RG*

Il s'agit de trouver toutes les paires $\rho =< condset, C_i >$, avec $condset$ une liste d'items et $C_i$ une catégorie, dont le support est supérieur au support minimum. Chaque paire $\rho$ correspond à une règle $condset \rightarrow C_i$ dont le support et la confiance sont définis par :

$$supp(\rho) = \frac{\text{\#textes de } C_i \text{ supportant condset}}{\text{\#textes de la base}}$$

$$conf(\rho) = \frac{\text{\#textes de } C_i \text{ supportant condset}}{\text{\#textes de la base supportant condset}}$$

Les paires dont le support est supérieur au support minimum sont des paires fréquentes. Si deux paires ont le même ensemble d'items, alors la paire ayant la confiance la plus élevée sera choisie comme règle. L'ensemble des règles d'association pour les catégories (CARs) est constitué de toutes les règles dont le support et la confiance sont supérieurs au support minimum et à la confiance minimum spécifiés par l'utilisateur.

Les motifs fréquents sont extraits en utilisant une unique valeur pour le support minimum quelle que soit la catégorie. Or, toutes les catégories ne contiennent pas le même nombre de documents. Un support minimum élevé ne permettra pas de trouver de motifs fréquents pour les petites catégories et à l'opposé, un support trop élevé va conduire à la génération d'un nombre trop important de règles pour les catégories contenant de nombreux textes. C'est pourquoi d'autres travaux ont proposé d'utiliser des valeurs de support minimum adaptées à chaque catégorie (msCBA) [JAN 03, LIU 00]. Les règles sont alors extraites en adoptant une stratégie de supports minimums multiples définis en adéquation avec la fréquence de distribution de chaque catégorie et un support minimum initial donné par l'utilisateur :

$$minSup_{C_i} = minSup_{intial} * frequenceDistribution(C_i)$$

avec, $frequenceDistribution(C_i) = \frac{\text{\#textes de } C_i}{\text{\#textes}}$.

3.1.2. *Le constructeur de classifieur CBA-CB*

Soit $R$ l'ensemble des règles CARs obtenues lors de l'étape précédente et $T_{Train}$ le jeu d'entraînement, le catégoriseur est construit à partir de la liste des règles $r_i \in R$ ordonnées suivant leur confiance. Chaque règle est ensuite testée sur $T_{Train}$. Si la règle n'améliore pas le taux de réussite du classifieur, alors la règle est éliminée de la liste des règles pour la catégorie examinée. L'algorithme 1 détaille le processus de construction du classifieur.

Le catégoriseur obtenu est une liste du type :

$< (r_1, r_2, ..., r_k), C_i >$ (où $C_i$ est la catégorie cible et $r_j$ une des règles associées)

Une fois le catégoriseur obtenu, pour tout nouveau texte à classer, les règles de classification sont évaluées sur le document tant qu'aucune règle n'est supportée. La catégorie affectée est alors la catégorie cible de la règle de classification ayant été validée pour le texte.

### 3.2. *Améliorations et autres approches*

Dans [JAN 03], les auteurs proposent de remplacer la mesure de confiance par l'intensité d'implication comme critère de tri des règles lors de la construction du classifieur. Dans [KUM 01], le classifieur est amélioré en lui adjoignant des arbres de décision afin d'améliorer le taux de réussite. Dans [ALI 97], les règles d'association sont utilisées pour permettre une classification partielle, c'est-à-dire que le système ne permet pas de classer pour toutes les catégories. En particulier, cette approche est intéressante dans le cas de valeurs manquantes. [BAR 03] propose également une méthode de classification basée sur les règles d'association mais contrairement à CBA-CB qui ne tient compte d'une seule règle pour prendre la décision d'affectation d'un texte à

**Data** : R : l'ensemble des règles CARS
D : le jeu d'entrainement
**begin**
$R$=tri(R);
**foreach** *règle* $r \in R$ **do**
$temp \leftarrow \emptyset$ ;
**foreach** *texte* $d \in D$ **do**
**if** $d$ *valide* $r$ **then**
$temp \leftarrow temp \bigcup d.id$ et marquer $r$ si $d$ a été bien classé ;
**end**
**end**
**if** $r$ *est marqué* **then**
ajouter $r$ dans $C$ ;
$D \leftarrow D \backslash temp$ ;
choisir une catégorie pour $C$ ;
calculer le taux d'erreur de $C$ ;
**end**
**end**
trouver la 1ère règle $p \in C$ minimisant le taux d'erreur et effacer toutes les règles après $p$;
retourner $C$;
**end**

**Algorithm 1**: Construire la catégoriseur CBA-CB

une catégorie, les auteurs proposent de considérer plusieurs règles puis d'adopter la catégorie majoritaire. Cette méthode intègre une étape d'élagage de règles basée sur le $\chi^2$, comme dans [LI 01]. De plus, un paramètre $maxrules$ définit le nombre maximal de règles à vérifier lors de l'étape de classification de nouveaux textes. Pour améliorer les performances, les règles sont classées par niveau, le système étudiera les règles du niveau supérieur si et seulement si aucune règle de classification n'a pu être mise en œuvre au niveau inférieur. Le système a été amélioré dans [BAR 04] en considérant une stratégie de supports minimums multiples.

Nous pouvons également citer d'autres travaux connexes. [HAD 03] propose une méthode basée sur les règles d'association dans le cadre de l'extraction de syntagmes nominaux. Dans ses travaux, l'auteur montre que les règles d'association sont très intéressantes pour rendre compte de la structure linguistique des textes. Cependant, les régles obtenues ne sont pas utilisées pour catégoriser les textes. [MAS 03a] propose une méthode de catégorisation pour l'analyse de comportements d'utilisateur de sites web (web usage mining) à l'aide de motifs séquentiels et de réseaux de neurones. Les motifs séquentiels sont utilisés pour diviser itérativement la base de données (logs) en sous-logs (eux-mêmes redivisés) représentant chacun un comportement différent. Les réseaux de neurones permettent de composer les groupes selon les motifs séquentiels trouvés. Cette méthode, bien qu'intéressante, n'est pas basée sur les motifs séquentiels

eux-mêmes pour l'étape de classification et n'est pas dédiée aux textes. L'utilisation des réseaux de neurones rend la méthode inapplicable face à de très gros volumes de données. De plus, la méthode est dédiée à l'analyse de comportements peu fréquents (pour ne pas découvrir de connaissance déjà connue) et n'est donc pas adaptée pour la classification d'un ensemble de données et la découverte de tendances. Enfin, il ne s'agit pas de classification supervisée mais plutôt de classification non-supervisée (hiérarchique, par divisions successives de la base).

En ce qui concerne les motifs séquentiels appliqués aux textes, nous pouvons citer [LEN 97, WON 00]. [WON 00] propose une approche intégrant deux méthodes. La première est basée sur la visualisation des occurrences des mots afin de détecter des motifs séquentiels. La seconde adopte un algorithme de recherche de motifs séquentiels.
Dans [LEN 97], les auteurs montrent l'intérêt d'utiliser les motifs séquentiels sur de grandes bases de documents, en particulier pour mettre en évidence les différentes tendances au cours du temps.

Comme nous avons pu le constater, la fouille de textes basée sur les motifs fréquents correspond soit à des travaux sur la classification à l'aide de règles d'association soit à d'autres problématiques résolues en adoptant les motifs séquentiels. Aucune approche à notre connaissance n'utilise les motifs séquentiels comme outil de classification de textes. Nous proposons donc une approche originale permettant d'intégrer une notion d'ordre au sein des textes tout en permettant le traitement de gros volumes de données.

Les sections suivantes présentent notre proposition et les expérimentations. Ces dernières détaillées section 5, soulignent de très bons résultats en terme de classification.

## 4. Catégorisation à base de motifs séquentiels : La méthode SPaC

Dans cette section nous présentons SPaC, une approche de catégorisation originale basée sur les motifs séquentiels. La méthode se décompose en deux phases. La première consiste à extraire un ensemble de motifs séquentiels à partir des textes du jeu d'entraînement. Puis la seconde génère un catégoriseur à l'aide des motifs séquentiels précédemment obtenus.

### 4.1. *Première étape - Des textes aux motifs séquentiels*

Chaque texte du jeu d'entraînement est transformé afin d'appliquer un algorithme de recherche de motifs séquentiels. En effet, les algorithmes utilisés sont basés sur une structuration des données de la forme <client, date, items>. Nous proposons de prendre en compte l'ordre des phrases au sein du texte mais les phrases quant à elles sont considérées comme des "sacs de mots". En effet, nous faisons l'hypothèse que

l'ordre des mots dans la phrase a une importance limitée mais que celui des phrases dans le texte a un impact lors du processus de classification (ceci est vérifié lors des expérimentations figure 1). Nous considérons donc chaque texte comme un client et chaque phrase comme une transaction estampillée par sa position au sein du texte. L'ensemble des mots représente l'ensemble des items et correspond à un itemset. Les mots d'une même phrase sont ainsi assimilés aux achats effectués par une client à une même date en adéquation avec la problématique du "panier de la ménagère" (section 2). Le tableau 2 résume les règles de correspondance entre les concepts associés aux motifs séquentiels et leur application aux données textuelles.

| **Base de données d'achats** | | **Base de données textuelle** |
|---|---|---|
| client | $\leftrightarrow$ | texte |
| item | $\leftrightarrow$ | mot |
| itemset/transaction | $\leftrightarrow$ | phrases (ensemble de mots) |
| date | $\leftrightarrow$ | position de la phrase dans le texte |

**Tableau 2.** *Correspondance entre les textes et les motifs séquentiels*

Nous réalisons un pré-traitement linguistique de type stemmatisation (radicalisation) et une suppression des mots non informatifs (stop-list) des textes traités. L'étape de stemmatisation supprime tous les suffixes des mots afin d'obtenir des stemmes qui sont plus génériques. Et l'étape de suppression des mots non informatifs élimine les mots comme "*le, des, une*" pouvant générer du bruit lors de la phase d'apprentissage. Toujours pour atténuer le bruit, une politique de suppression des mots non discriminants, basée sur une mesure d'entropie, a été mise en œuvre. Ceci nous permet d'effectuer une recherche de motifs avec un plus faible support sans générer un trop grand nombre de candidats.

L'élimination par entropie est réalisée sur la base d'un seuil. Pour chaque mot $w$, $H(w)$ définit l'entropie de ce mot pour l'ensemble des catégories $C_i$ :

$$H(w) = -\sum_{C_i} p(w).p(C_i|w).log(p(C_i|w)) + ((1-p(w)).p(C_i|\bar{w}).log(p(C_i|\bar{w})))$$

Le seuil d'élimination a été déterminé de manière empirique. Les meilleurs résultats ont été obtenus en éliminant 5 à 10% des mots. Ces résultats concordent avec ceux définis par la loi de Zipf [SAL 75]. Dans la suite de cet article, nous utiliserons les notations introduites tableau 3.

SPaC extrait l'ensemble des motifs séquentiels selon une politique de supports minimums multiples identique à celle de msCBA. Cela permet de définir un support pour chacune des catégories $C_i$. Lors d'une recherche de motifs séquentiels avec un support de 10%, la recherche ne s'effectuera plus sur toute la base mais catégorie par catégorie. Ainsi, les motifs séquentiels d'une catégorie contenant peu de textes

| Notation | Signification |
|---|---|
| $\mathcal{C} = \{C_1, \ldots, C_n\}$ | l'ensemble des $n$ catégories |
| $C_i \in \mathcal{C}$ | une catégorie de C |
| $minSup_{C_i}$ | le support minimum de la catégorie $C_i$, défini par l'utilisateur |
| $T$ | l'ensemble des textes |
| $T^{C_i} \subseteq T$ | les textes appartenant à $C_i$ |
| $T_{Train} = \{(C_i, T^{C_i})\}$ | le jeu d'apprentissage constitué d'un ensemble de textes associés à leur catégorie. |
| $SEQ$ | accesseur contenant l'ensemble des séquences ordonnées par catégorie $C_i$, client $c$ et date $t$ |
| $SP$ | un tableau de motifs séquentiels |
| $RuleSP$ | un tableau de n-uplets $(sp_j, C_i, conf_{i,j})$ correspondant à la séquence $sp_j$, la catégorie $C_i$ et la confiance $conf_{i,j}$ de la règle $sp_j \rightarrow C_i$ |

**Tableau 3.** *Notations*

(dont le nombre est inférieur à 10% de la base) ne seront pas ignorés. Contrairement à msCBA où le support minimum est défini selon une formule (Section 3.1.1), dans SPaC, les supports minimums de chaque catégorie sont définis par l'utilisateur. Ceci permet d'affiner l'étape de classification en intégrant dans le processus la connaissance des experts pour chaque catégorie. Les expérimentations réalisées dans [BAR 04] indiquent l'importance d'un support spécifique à chaque classe et au sein d'un même jeu de données, les supports optimaux entre les différentes classes varient nettement. Nous divisons le jeu d'entraînement en $n$ sous-ensembles correspondant aux textes des $n$ catégories. Ensuite, un algorithme de recherche de motifs séquentiels est appliqué sur chacun des sous-ensembles selon le support minimum spécifié. Les motifs séquentiels fréquents sont donc obtenus pour chaque catégorie et leur support conservé. Le support d'un motif fréquent correspond au nombre de textes qui le supporte (ou qui le contient).

**Definition 1** *Soit $< s_1 \ldots s_p >$ une séquence. Le support de $< s_1 \ldots s_p >$ est défini par :*

$$supp(< s_1 \ldots s_p >) = \frac{\#\textit{textes supportant } < s_1 \ldots s_p >}{\#\textit{textes de la base}}$$

L'algorithme 2 définit la phase d'extraction de motifs séquentiels. La fonction $SPMining()$ appelle l'algorithme SPAM [AYR 02] pour rechercher les séquences fréquentes.

**Data** : $T_{Train}$ : jeu d'entraînement
$\{minSup_{C_i}\}$ : l'ensemble des supports minimums pour chacune des catégories $C_i$
**Result** : SP : un ensemble de motifs séquentiels
**begin**
$SEQ \leftarrow \emptyset$ ; $customer \leftarrow 0$ ; $timestamp \leftarrow 0$;
**foreach** *Catégorie* $C_i \in \mathcal{C}$ **do**
**foreach** *Texte* $T_j \in T^{C_i}$ **do**
**foreach** *Phrase* $S_k \in T_j$ **do**
$V_s$=TFIDF( Stemme($S_k$)) ; // Génère un vecteur de type $TF$-$IDF$ à partir de la phrase $S_k$
**for** *(*$s = 0; s <\mid V_s \mid; s++$*)* **do**
**if** $V_s[s] > 0$ **then**
$SEQ[C_i][customer][timestamp]$.additem(s);
**end**
**end**
timestamp++;
**end**
$timestamp \leftarrow 0; customer++$;
**end**
$customer \leftarrow 0$;
**end**
**foreach** *Categorie* $C_i \in C$ **do**
SP[$C_i$]=SPMining(SEQ[$C_i$],$minSup_{C_i}$);
**end**
**end**

**Algorithm 2**: $SPaC - RG$ : génération des règles

Par exemple, les motifs fréquents suivants ont été extraits de la catégorie "Achats-Logistique" du jeu de données français :

< (cacao) (ivoir) (abidjan)>
< (blé soj) (mai)>
< (soj)(blé lespin victor)(maï soj )(maï )(grain soj)(soj tourteau )>

Le premier motif indique que pour les textes de la catégorie, il apparaît régulièrement une phrase contenant le mot *cacao* suivie d'une phrase contenant le mot *ivoire* et enfin une phrase contenant *Abidjan*. Le second motif séquentiel signifie qu'un certain nombre de textes (au moins le support minimal) contiennentt les mots *ble* et *soja* au sein d'une même phrase suivie d'une phrase contenant *maï*. Le troisième motif séquentiel peut être interprété par exemple : le mot *maï* apparaît dans deux phrases successives et est suivi par une phrase contenant le mot *grain*.

L'utilisation de motifs séquentiels permet de prendre en compte certaines occurrences multiples de mots (contrairement aux règles d'associations).

### 4.2. *Deuxième étape - Des motifs séquentiels aux catégories*

L'objectif de cette seconde étape est de générer un catégoriseur à partir des motifs séquentiels extraits lors de l'étape précédente. Cette construction est basée sur une notion de confiance et se définit comme suit : Pour chaque motif séquentiel $< s_1 \dots s_p >$ extrait pour une catégorie $C_i$, la règle $\gamma$ est définie de la façon suivante :
$\gamma :< s_1 \dots s_p > \rightarrow C_i$. Cette règle signifie : si un texte contient $s_1$ suivi de $s_2 \dots$ et de $s_p$, alors le texte valide son appartenance à $C_i$. La confiance de cette validation est déterminée par la confiance de la règle définie par :

$$conf(\gamma) = \frac{\text{\#textes de } C_i \text{ supportant} <s_1 \dots s_p>}{\text{\#textes de la base supportant} <s_1 \dots s_p>}$$

Plus la confiance d'une règle est grande, plus le motif séquentiel est discriminant pour la catégorie qui lui est associée. Les règles sont ensuite ordonnées selon leur confiance et selon la taille de leur séquence (second critère).

Pour chaque nouveau texte à classer, la politique de catégorisation est la suivante : Une fois la liste des règles ordonnée, on parcourt cette liste de façon décroissante en appliquant les motifs séquentiels de chacune des règles au texte à catégoriser. Une fois les $K$ premières règles valides trouvées, le texte est affecté à la catégorie majoritaire défini sur ces $K$ règles. Cette méthode correspond à la méthode de "vote majoritaire" adoptée dans [BAR 03]. Si deux ou plusieurs catégories obtiennent le même score, alors un choix aléatoire est effectué pour déterminer la catégorie d'appartenance du texte. Il se peut qu'il n'existe pas $K$ règles valides. Dans ce cas particulier, le vote majoritaire s'effectue normalement sur les $n$ règles valides (avec $n$<$K$). Toutes les règles ont le même poids dans ce vote. Et si finalement il n'existe aucune règle valide pour le texte en question, alors ce dernier n'est pas catégorisé.

L'étape de catégorisation de SPaC est décrite par l'algorithme (SPaC-C) suivant :

**Data** : $T_{Test}$ : A Test Set
KFS (le paramètre $K$),
SP (le tableau des motifs séquentiels généré par SPaC-RG)
**begin**
$nb \leftarrow 1$ ;
**foreach** *Catégorie* $C_i \in C$ **do**
**foreach** $sp_j \in$ *SP[*$C_i$*]* **do**
$RuleSP[nb] \leftarrow (sp_j, C_i, conf(sp_j \rightarrow C_i)$ ;
$nb++$ ;
**end**
**end**
Trier $RuleSP$ selon la confiance des règles (et par la taille de la séquence en second critère);
$nfs \leftarrow 0$ ; $classable \leftarrow 0$;
**foreach** *Texte* $T_k \in T_{Test}$ **do**
**foreach** *Règle* ($sp_j \rightarrow C_i$)$\in RuleSP$ **do**
**if** $T_k$ *supporte* $SP_j$ **then**
$T_k$.score[$C_i$]++ ; $classable \leftarrow 1$ ; $nfs$++ ;
**if** $nfs \geq KFS$ **then** break // (Sort de la boucle "foreach Règle".)
**end**
**end**
**if** *classable* **then**
Affecter $T_k$ à la catégorie ayant obtenu le meilleur score;
**end**
classable $\leftarrow 0$ ; nfs $\leftarrow 0$ ;
**end**
**end**

**Algorithm 3**: SPaC-C : étape de catégorisation

## 5. Expériences

Les expériences ont été réalisées sur trois bases de données (corpus). Les deux premières bases "20 Newsgroups" et "Reuters" [HET 99] sont deux corpus anglophones de référence en catégorisation automatique de texte. Pour la troisième base, nous avons utilisé un corpus français de dépêches d'agences. Cette dernière base est constituée de $8239$ textes répartis en $28$ categories.

Ces trois différents corpus ont été catégorisés selon CBA et SVM puis à l'aide du catégoriseur SPaC. Les tableaux 4, 5 et 6 détaillent les résultats obtenus selon les différents corpus et les différentes méthodes. Pour permettre une comparaison plus précise, nous évaluons les processus de catégorisation à l'aide de la mesure $F_\beta$ [RIJ 79]. Cette mesure permet de combiner rappel et précision afin de fournir une note globale pour la méthode. Il s'agit d'une mesure plus équilibrée que le taux de réussite qui favorise le silence du classifieur. Le taux de réussite est la mesure adoptée par les approches basées sur les motifs fréquents [LIU 98, BAR 03] alors que la majorité des

méthodes de classification utilisent des mesures basées sur le rappel et la précision [JéR 03, SEB 02b]

**Definition 2** *La mesure $F_\beta$ se définit par :*

$$F_\beta = \frac{(\beta^2 + 1)\pi_i \rho_i}{\beta^2 \pi_i + \rho_i}$$

*Où $\rho_i$ et $\pi_i$ représentent respectivement le rappel et la précision obtenues pour une catégorie $C_i$.*

**Definition 3** *Précision et rappel :*

$$\pi_i = \frac{VP_i}{VP_i + FP_i} \quad , \quad \rho_i = \frac{VP_i}{VP_i + FN_i}$$

*où $VP_i$, $FP_i$, $FN_i$ définissent respectivement les textes bien classés (Vrai Positif), les textes assignés par erreur (Faux Positifs) ainsi que les textes omis par le catégoriseur (Faux Négatif) pour une catégorie $C_i$.*

Le taux de réussite quant à lui est défini par :

**Definition 4** *Taux de réussite (Accuracy rate) :*

$$A_i = \frac{VP_i + VN_i}{VP_i + VN_i + FP_i + FN_i}$$

La bonification au silence du classifieur est réalisée en tenant compte de $VN_i$ (Vrai Négatif) c'est-à-dire le texte n'appartient pas à la catégorie et n'a pas été assigné à la catégorie par le système.

Par la suite, nous utiliserons les mesures de micro-moyenne et de macro-moyenne pour évaluer le catégoriseur dans sa globalité [SEB 02b, YAN 99].

**Definition 5** *Micro-moyenne ($\mu$) et Macro-moyenne ( M) :*

$$\hat{\pi}^\mu = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|}(VP_i + FP_i)}, \hat{\rho}^\mu = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|}(VP_i + FN_i)} \hat{\pi}^M = \frac{\sum_{i=1}^{|C|} \hat{\pi}_i}{|C|}, \hat{\rho}^M = \frac{\sum_{i=1}^{|C|} \hat{\rho}_i}{|C|}$$

La micro-moyenne accorde la même importance à la performance de chaque document contrairement à la macro-moyenne qui réalise une moyenne par catégorie.

| | SPaC | msCBA | SVM |
|---|---|---|---|
| $F_1$ macro-averaging | 0.444 | 0.367 | **0.485** |
| $F_1$ micro-averaging | **0.487** | 0.401 | 0.486 |
| Taux de réussite | 0.962 | 0.956 | **0.969** |
| Paramètres | multisupport = 3% | multisupport = 0.7% | C = 1 |

**Tableau 4.** *Comparaison de SPaC, msCBA et SVM sur le jeu de données français avec un jeu d'entraînement de 33%*

Dans toutes les expériences, nous avons utilisé $\beta = 1$ pour la mesure $F_\beta$. De cette manière une importance identique est accordée au rappel et à la précision

Pour la méthode des SVM, nous avons considéré un noyau linéaire à l'aide du package SVM$^{Light}$ [JOA 98a] avec comme paramètre c=1 (aucune amélioration n'a été obtenue avec des noyaux non linéaires). Les supports choisis sont ceux qui ont permis d'obtenir les meilleurs résultats tout en restant calculables. En effet, des supports trop bas peuvent engendrer un nombre trop important de candidats. SPaC a été utilisé avec le paramètre $K = 10$. C'est-à-dire au plus 10 règles sont prises en compte pour catégoriser un texte.

L'intérêt de SPaC est d'utiliser des connaissances compréhensibles et exploitables par l'utilisateur. Ces informations sous forme de motifs séquentiels permettent de mettre en évidence des tendances et de comprendre réellement ce qui discrimine les catégories. Il apparaît que l'information extraite grâce aux motifs séquentiels est au moins aussi importante que le fait d'obtenir le "meilleur" des catégoriseurs. Pour cette raison, la comparaison est essentiellement étudiée entre CBA et SPaC. Dans les expériences, CBA est testé dans sa version multi-support (msCBA). Les supports utilisés ont été ceux permettant d'obtenir les meilleurs résultats de catégorisation. Les résultats montrent que SPaC est toujours meilleur que msCBA en macro-moyenne. Ceci est dû au fait que SPaC obtient des résultats plus uniformes sur l'ensemble des catégories contrairement à CBA qui tend à maximiser les résultats des catégories majoritaires (notamment dans Reuters où 2 catégories (sur 90) représentent 2/3 du jeu de test). SPaC est donc plus robuste sur les catégories minoritaires dont l'apprentissage est parfois plus complexe. Les tableaux 4, 5 et 6 montrent que SPaC est dans l'ensemble plus performant que CBA. Et pour ce qui concerne l'approche à base de SVMs, SPaC obtient des performances identiques en français et supérieures sur le corpus "20 Newsgroups". Néanmoins, la base de données"Reuters" reste indéniablement favorable à SVM.

Figure 1, nous analysons les variations obtenues en $F_1$ lors d'une augmentation du paramètre $K$ de 1 à 15. L'expérience montre qu'avec $K = 10$, les résultats sont pratiquement maximaux (ce résultat reste similaire avec [BAR 03] qui, lui, définit son seuil à $maxrules = 9$). De plus, les expériences ont été réalisées (1) avec ordre : le motif séquentiel (séquence) est supporté par le texte (2) sans ordre : les sous séquences

| | SPaC | msCBA | SVM |
|---|---|---|---|
| $F_1$ macro-averaging | 0.219 | 0.082 | **0.500** |
| $F_1$ micro-averaging | 0.591 | 0.679 | **0.840** |
| Taux de réussite | 0.990 | 0.992 | **0.996** |
| Paramètres | multisupport différent selon la categorie | multisupport = 1% | C = 1 |

**Tableau 5.** *Comparaison de SPaC, msCBA and SVM sur le jeu de données Reuters*

| | SPaC | msCBA | SVM | |
|---|---|---|---|---|
| $F_1$ macro-averaging | 0.400 | **0.463** | 0.423 | 0.423 |
| $F_1$ micro-averaging | 0.435 | **0.502** | 0.436 | 0.455 |
| Taux de réussite | **0.947** | 0.941 | 0.941 | |
| Paramètres | multisupport = 0.03 | multisupport 0.035 | C=1 | |

**Tableau 6.** *Comparison de SPaC, msCBA and SVM sur le jeu de données "20 Newsgroups" avec un jeu d'entraînement de 33%*

de la séquence sont supportées par le texte sans ordre défini. Les sous-séquences sont chacune supportées par une phrase du document, mais l'ensemble des phrases n'est pas forcement celui indiqué par le motif séquentiel. Les expériences montrent que lorsque l'ordre de la séquence est respecté au sein du document, les résultats sont meilleurs.

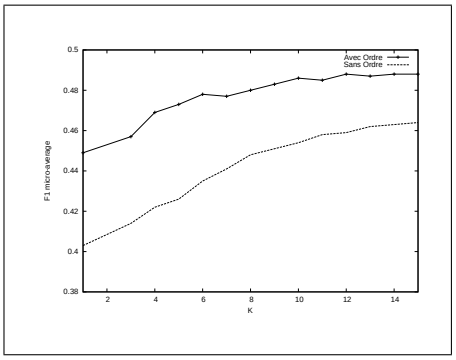


**Figure 1.** *Variation de $F1$ en fonction du paramètre $K$*

## 6. Conclusion

Dans cet article, nous avons proposé un système de catégorisation basé sur les motifs séquentiels. L'extraction des motifs séquentiels est réalisée pour chaque catégorie à partir d'une représentation textuelle des textes de type $TF$-$IDF$ et d'une transformation en terme de triplet <client,date,item>. Dans cette approche, chaque nouveau texte à classer est affecté à une catégorie en fonction des différents motifs séquentiels qu'il supporte grâce à une politique de "vote majoritaire".

SPAC obtient des résultats meilleurs que msCBA et atteint ceux des SVM pour des bases difficiles. Toutefois, SVM reste le plus efficace sur le corpus Reuters. Néamoins, il est important de souligner que l'extraction de connaissances "compréhensibles" est un atout tout aussi important qu'une bonne catégorisation. Ces descriptions sont primordiales pour les experts démunis face aux grandes quantités de textes à analyser et traiter. Les motifs séquentiels peuvent être aisément analysés afin de mieux comprendre les forces et les faiblesses du catégoriseur construit et peuvent aussi être utilisés pour rechercher les tendances au sein des bases de textes. Dans notre problématique de catégorisation, les motifs séquentiels ont montré un potentiel très attractif. De plus, SPaC est efficace lorsque des catégoriseurs reconnus comme SVM montrent leurs limites, comme par exemple sur la base "20 Newsgroups".

Différentes voies d'amélioration sont envisagées. Tout d'abord, l'utilisation des motifs séquentiels généralisés [AGR 95] incluant les contraintes de temps permettrait d'améliorer les performances. La mise en œuvre de différents niveaux de motifs comme proposés dans [BAR 04] permet de conserver des règles de classification très spécifiques sans nuire aux performances générales du classifieur. Il s'agit d'utiliser un ensemble de règles compactes afin de diminuer le support minimum. Dans cet objectif, il serait intéressant d'étendre les travaux de [CRE 02] aux motifs séquentiels : utiliser les $\delta-$libres pour trouver des règles de classification dont la partie gauche est la plus courte possible.

Un de nos objectifs est de proposer un catégoriseur incrémental. En effet, il est possible de mettre à jour les motifs séquentiels extraits sans ré-exécuter tout le processus d'extraction [MAS 03b]. Ceci est très important pour la catégorisation automatique de nouvelles où l'actualité est sans cesse mouvante. Les motifs séquentiels sont donc, à notre avis, un premier pas vers la classification temps réel de textes, ou On-Line Text Classification Process (OLTCP).

## 7. Bibliographie


[AGR 94] AGRAWAL R., SRIKANT R., « Fast Algorithms for Mining Generalized Association Rules », *Proc. of the 20th Int. Conf. on Very Large Databases (VLDB'94)*, September 1994.

[AGR 95] AGRAWAL R., SRIKANT R., « Mining sequential patterns », *Eleventh Int. Conf. on Data Engineering*, IEEE Computer Society Press, 1995, p. 3–14.

[AGR 00] AGRAWAL R., JR. R. J. B., SRIKANT R., « Athena : Mining-Based Interactive Management of Text Databases », *Extending Database Technology*, 2000, p. 365-379.

[ALI 97] ALI K., MANGANARIS S., SRIKANT R., « Partial Classification Using Association Rules », *Knowledge Discovery and Data Mining*, 1997, p. 115-118.

[AYR 02] AYRES J., GEHRKE J., YIU T., FLANNICK J., « Sequential Pattern Mining Using Bitmaps », *Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.*, 2002.

[BAR 03] BARALIS E., GARZA P., « Majority Classification by Means of Association Rules », *7th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2003, p. 35-46.

[BAR 04] BARALIS E., CHIUSANO S., GARZA P., « On support thresholds in associative classification », *Proc. of the 2004 ACM Symposium on Applied Computing (SAC)*, 2004, p. 553-558.

[CRE 02] CREMILLEUX B., BOULICAUT J., « Simplest rules characterizing classes generated by delta-free sets », *Proceedings of the 22nd BCS SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence ES 2002*, Springer-Verlag, 2002, p. 33–46.

[HAD 03] HADDAD H., « Utilisation des syntagmes Nominaux dans un système de recherche d'information », *Actes des 19 èmes Journées Bases de Données Avancées (BDA'03)*, 2003, p. 129–145.

[HET 99] HETTICH S., S.D.BAY, *The UCI KDD Archive. [http ://kdd.ics.uci.edu]*, Irvine, CA : University of California, Department of Information and Computer Science., 1999.

[IWA 95] IWAYAMA M., TOKUNAGA T., « Cluster-based text categorization : a comparison of category search strategies », *Proc. of SIGIR-95, 18th ACM Int. Conf. on Research and Development in Information Retrieval*, ACM Press, 1995, p. 273–281.

[JAN 03] JANSSENS D., WETS G., BRIJS T., VANHOOF K., G. C., « Adapting the CBA-algorithm by means of intensity of implication », *Proc. of the First Int. Conf. on Fuzzy Information Processing Theories and Applications*, 2003, p. 397-403.

[JOA 98a] JOACHIMS T., « Making large-scale support vector machine learning practical », B. SCHOLKOPF C. BURGES A. S., Ed., *Advances in Kernel Methods : Support Vector Machines*, MIT Press, Cambridge, MA, 1998.

[JOA 98b] JOACHIMS T., « Text categorization with support vector machines : learning with many relevant features », *Proc. of ECML-98, 10th European Conf. on Machine Learning*, Chemnitz, DE, 1998, Springer Verlag, Heidelberg, DE, p. 137–142.

[JéR 03] JÉROME AUGÉ KURT ENGLMEIER G. H., MOTHE J., « Catégorisation automatique de textes basée sur des hiérarchies de concepts », *BDA'03 Journées Bases de données avancées*, 2003, p. 69-87.

[KUM 01] KUMAR V., ET AL, Eds., *Classification Using Association Rules : Weaknesses and Enhancements*, 2001.

[LEN 97] LENT B., AGRAWAL R., SRIKANT R., « Discovering Trends in Text Databases », *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, AAAI Press, 14–17 1997, p. 227–230.

[LI 01] LI W., HAN J., PEI J., « CMAR : Accurate and Efficient Classification Based on Multiple Class-Association Rules », *Proc. 2001 Int. Conf. on Data Mining (ICDM'01)*, 2001.

[LIU 98] LIU B., HSU W., MA Y., « Integrating Classification and Association Rule Mining », *Knowledge Discovery and Data Mining*, 1998, p. 80-86.

[LIU 00] LIU B., MA Y., WONG C. K., « Improving an Association Rule Based Classifier », *Principles of Data Mining and Knowledge Discovery*, 2000, p. 504-509.

[MAR 61] MARON M., « Automatic indexing : An experimental inquiry », *Journal of the ACM (JACM)*, vol. 8, 1961, p. 404–417.

[MAS 03a] MASSEGLIA F., « Diviser pour découvrir : une méthode d'analyse du comportement de tous les utilisateurs d'un site web », *Actes des 19 èmes Journées Bases de Données Avancées (BDA'03)*, 2003, p. 227–246.

[MAS 03b] MASSEGLIA F., PONCELET P., TEISSEIRE M., « Incremental Mining of Sequential Patterns in Large Databases », *Data and Knowledge Engineering*, vol. 46, n° 1, 2003.

[RIJ 79] RIJSBERGEN C. J. V., *Information Retrieval*, Butterworths, sec. edition, 1979.

[SAL 75] SALTON G., YANG C., YU C., « A theory of term importance in automatic text analysis », *Journal of the American Society for Information Science*, vol. 36, 1975, p. 33–44.

[SAL 83] SALTON G., MCGILL M. J., *Introduction to modern information retrieval*, 1983.

[SEB 02a] SEBASTIANI F., « Machine Learning in Automated Text Categorisation », *Proc. of ACM Computing Surveys*, vol. 34, 2002, p. 1–47.

[SEB 02b] SEBASTIANI F., « Machine learning in automated text categorization. », *ACM Computing Surveys*, vol. 34, n° 1, 2002, p. 1-47.

[SRI 96] SRIKANT R., AGRAWAL R., « Mining Sequential Patterns : Generalizations and Performance Improvements », *Proc. of the 5th Int.Conf. on Extending Database Technology (EDBT'96)*, September 1996, p. 3-17.

[VAP 95] VAPNIK S. N., *The Nature of Statistical Learning Theory*, Springer, 1995.

[WAN 00] WANG K., ZHOU S., HE Y., « Growing decision trees on support-less association rules », *Knowledge Discovery and Data Mining*, 2000, p. 265-269.

[WON 00] WONG P. C., COWLEY W., FOOTE H., JURRUS E., THOMAS J., « Visualizing Sequential Patterns for Text Mining », *INFOVIS*, 2000, p. 105-114.

[YAN 99] YANG Y., « An evaluation of statistical approaches to text categorization », *Information Retrieval Journal*, vol. 1, 1999, p. 69-90.