Chapitre 6

Classification Automatique des textes : Approche Orientée Agent

Table des matières

6.1- Introduction

Après l'étude faite sur l'état de l'art des différentes approches pour la construction de modèles de catégorisation de textes ainsi que les divers codages et techniques pour la représentation des textes, développés dans la littérature, nous avons adoptés pour nos travaux :

La méthode des N-grams pour représenter notre corpus pour son indépendance des différentes langues et son non exigence des traitements linguistiques préalables et d'autres avantages qui vont être décrits dans la section suivante.

Pour l'algorithme d'apprentissage et classification, le modèle d'indépendance conditionnelle (Naïve Bayes classifier) a été utilisé pour sa simplicité d'une part, et d'autre part, comme tous les modèles probabilistes, il s'appuie sur une base théorique précise.

Pour pouvoir confronter les différents résultats fournis par les classifieurs construits classiques ou Muti-Agents, nous allons utiliser les mesures de rappel et précision ainsi que la F-measure (F_1) pour évaluer les performances de ces modèles.

Pour la partie applicative, au lieu de procéder par une démarche classique, nous proposons de distribuer l'intelligence dans une approche qui consiste à déléguer la tâche de classification de textes à un Système Multi-Agents.

Ce chapitre va contenir le produit fini de nos travaux de recherche. Nous commençons par décrire notre approche d'une façon générale, ensuite nous allons motiver et justifier tous les choix des solutions adoptées durant toutes les phases du processus, en commençant par la méthode de représentation de textes choisie, suivie par l'algorithme d'apprentissage et classification utilisée qui va être détaillé, ainsi que l'intérêt de solliciter un SMA pour un tel traitement. La base de texte utilisé pour l'apprentissage et la classification sera décrite par la suite avant d'enchainer par une présentation des résultats expérimentaux et l'influence de quelques facteurs dans ces résultats et nous terminons par discuter les résultats obtenus et une conclusion.

6.2- Description générale de l'approche

Dans le but de présenter à l'utilisateur une application performante, en matière de qualité de résultats et rapidité d'exécution, il est intéressant de distribuer le processus de classification en définissant une architecture composée de parties distinctes, chaque partie va traiter le problème de catégorisation de textes d'une façon spécifique, mais pouvant communiquer pour partager leurs connaissances.

L'idée générale de la présente architecture se présente de la manière suivante :

Notre système de catégorisation de textes va être composé de plusieurs modules logiciels (Agent), le nombre d'agents va être fixé après l'expérimentation et évaluation qui va solliciter 3, 9, 21, 33, 61, 99 et 181 agents, après une étude qui va porter sur le compromis qualité-des résultats, efficacité en termes de temps d'exécution. Deux types d'agents délibératifs (cognitifs), vont être utilisés, des agents classifieurs pour catégoriser les documents en première manche et un agent administrateur central qui va recueillir tous les suffrages pour consolider les votes et éventuellement trancher pour la catégorie qui va être élue.
Dans nos travaux, nous allons varier le "N" des N-grammes, de même pour le nombre d'agents qui va être renforcé au fur et à mesure, jusqu'à stabilisation des résultats en matière de qualité et de rapidité, à chaque variation de paramètres les mesures d'évaluation de performances rappel, précision et F-measure (F_1) seront nos marques pour jauger nos classifieurs.

- Dans la phase d'apprentissage chaque classe va être répartie sur le nombre d'agents choisis. Par conséquent notre base de d'apprentissage est scindée en plusieurs sous bases secondaires réduites en nombre de textes par classe. Chaque base secondaire contiendra un échantillon de toutes les classes de la base principale Reuters. Chaque agent possèdera sa propre base sur laquelle il exercera son apprentissage.
  Comme on a vu précédemment dans la section 3.2.8, les différents classifieurs peuvent correspondre à différents algorithmes ou au même algorithme utilisé avec différents sous-échantillons du corpus d'apprentissage, notre approche se situe dans le deuxième cas qui va procéder avec le même algorithme de classification « Naïve Bayes », avec différents mini-corpus d'apprentissage.
- Dés que l'apprentissage soit terminé, la phase de classification des documents test est lancée : Chaque texte va être traité par tous les agents qui utilisent le modèle Naïve Bayes, une probabilité va être accordée pour l'appartenance du texte à chaque classe, le texte sera catégorisé dans la classe qui possède la plus grande probabilité.
  Chaque agent dans le système fera sa propre classification pour le même texte (Autonomie).
  La décision finale sera prise après un vote majoritaire (Collaboration), le texte sera catégorisé dans la classe qui a été nommé par le plus grand nombre d'agents, et puisque le nombre d'agents a été choisi impair volontairement, l'égalité parfaite dans le vote est évitée, et une catégorie prendra toujours l'ascendant sur les autres.
- Après une étude portée sur les résultats fournis par les différents modèles de classification construits, l'ensemble des paramètres (n-grammes et nombre d'agents) sera fixé pour le modèle adopté.
- Le modèle final accepté va servir à classer de nouveaux textes de catégorie inconnue de la même manière de classification des documents test.

6.3- Motivations

6.3.1- Codage en n-grammes

Comme on a vu dans les chapitres précédents, la première phase dans le processus de classification de textes, est de subdiviser le texte à traiter en plusieurs unités d'information qu'on peut appeler termes ou descripteurs qui sont, habituellement, des mots simples. La question principale qui se pose dans cette première phase de préparation des documents du corpus est : Sur le plan informatique, comment repérer un mot ? D'une autre manière, quels sont les bornes formelles pour délimiter un mot ? Si pour les langues comme le français et l'anglais la réponse est presque évidente – à savoir que toute chaine de caractères précédée et suivie d'un espace ou un signe de ponctuation est considérée comme un mot simple- Cette règle n'est pas valable pour les autres langues.

Si le mot simple ne convient pas à toutes les langues, quelle est donc l'unité d'information élémentaire la plus appropriée pour fractionner un document ?

Contrairement à la recherche d'information, dans un contexte de classification de textes, le fait que les unités d'information élémentaires n'ont pas vraiment un sens, n'est pas une contrainte lors de la classification, néanmoins ces unités atomiques doivent être facilement identifiées sur le plan informatique, et peuvent être comparées statistiquement.

Si on considère le mot comme unité d'information de base, une classification mutilingue est impossible. En traitant les mots comme des termes, la représentation des textes s'avère relativement simple pour le français ou l'anglais, mais très difficile pour des langues comme

l'allemand ou l'arabe. D'autre part, le stemming et la lemmatisation utilisés comme moyen de normalisation et de réduction du lexique constitue une contrainte non moins négligeable. La notion de n-grammes, qui depuis une quinzaine d'années donne de bons résultats, est devenue, par des récentes recherches, un axe privilégié, dans le domaine de classification de textes.

Toutes ces raisons ont appuyé notre choix, pour la suite du travail qu'on va accomplir pour le codage et la représentation des documents, sur les techniques basées sur les n-grammes qui garantissent plusieurs avantages, confirmés par plusieurs auteurs, dont le principaux sont les suivants :

- La représentation en n-grammes s'attaque aux documents presque dans leur état brut, contrairement aux autres représentations qui nécessitent des traitements purement linguistiques, comme les traitements d'élimination des mots vides, de stemming et de lemmatisation (Jalam, 2003). Ces traitements améliorent la performance des systèmes à base sur les mots mais en contrepartie la mise en ouvre informatique de ces procédures est relativement lourde. D'autre part, l'étude de (Sahami, 1999) a montré que la performance des systèmes à base des n-grammes ne progresse pas même après ces traitements linguistiques, Et ce ci peut être confirmé par le fait que si un texte contient plusieurs mots de même racine, le nombre des n-grammes correspondants augmentera sans le moindre traitement linguistique préalable (Jalam, 2003).
- Un autre point à souligner en faveur des n-grammes : quelques algorithmes de lemmatisation ne semble pas être en mesure de regrouper des termes comme *automatisation***,** *automatiser* et *automatique* dans la même classe, par contre, le découpage en n-grammes est suffisant pour classer les trois termes dans la même classe, les tri-grams : *aut, uto, tom, oma, mat, ati,* permettent par une mesure de similarité d'affirmer que c'est l'*automatique* dont il est question.
- Le codage en n-grammes n'a pas besoin de segmenter le texte avant d'extraire les termes, ce qui offre à cette technique une caractéristique très intéressante, capable de représenter et traiter les langues pour lesquelles les frontières entre termes ne sont pas bien marquées comme l'allemand, le chinois, ou la langue arabe, dans laquelle les pronoms, sujets et compléments sont liés dé fois aux verbes, une seule chaîne de caractères unie représentant une phrase comme, par exemple, « kalamtouhou » ("je lui ai parlé") (Jalam, 2003). Notons bien, qu'une segmentation en n-grammes de caractères satisfait toutes les langues qui utilisent un alphabet, pour reconstruire le texte on procède à la concaténation, (Biskri & Delisle, 2001).
- Le choix des n-grams permet aussi de contrôler la taille du lexique et de le conserver à un seuil raisonnable. Parmi les grandes difficultés, qui s'opposent aux algorithmes d'analyse des grands corpus, c'est la taille du lexique. En effet, un découpage en mots fait que la taille du lexique, est d'autant plus importante, que le corpus est important. Cette contrainte persiste, malgré les aménagements appliqués sur les textes durant la phase de prétraitement (Lemmatisation, suppression des mots-vides, etc..). Le nombre de n-grammes d'un corpus, ne peut dépasser la taille de l'alphabet à la puissance n. Un découpage en quadri-grams pour la langue anglaise (Alphabet de 26 caractères) nous donne une taille maximale de 26^4 entrées, soit un vocabulaire d e 456 976 qaudri-grammes possibles. Si on élimine les combinaisons comme FFFF, RRRM, KPPP, etc.., qu'on ne trouvera jamais, ce nombre diminue d'une façon considérable. (Lelu & Hallab ,2000) estime ce nombre à quelques 13 087 quadri-grams pour un texte de 173 000 caractères.

- Ce type de codage est multilingues : La langue du document ne pose aucune contrainte particulière à sa représentation en n-grammes. En conséquence, aucune connaissance linguistique préalable n'est requise, contrairement aux systèmes basés sur les mots qui sont dépendants des langues dans lesquels il faut utiliser des dictionnaires spécifiques à chaque langue (féminin-masculin ; singulier-pluriel ; conjugaisons ; etc.) (Jalam, 2003),(Clech, 2004).
- Comparativement à d'autres techniques, les n-grammes extraient automatiquement les racines des mots les plus fréquents (Jalam, 2003),(Clech, 2004)
- Enfin, les erreurs d'orthographe et les éventuelles déformations d'un texte relatives à l'utilisation des systèmes de reconnaissance optique de caractères (OCR) n'ont pas d'incidence grave sur le profil d'un document. La reconnaissance optique d'un texte scanné est en général approximative. Par exemple, le mot "chapitre" peut être lu comme "clapitre". Une représentation à base de mots aura du mal à reconnaître qu'il s'agit du mot "chapitre" puisque le mot est mal orthographié, tandis qu'une représentation à base des n-grammes capture normalement les autres n-grammes significatives comme "apit", "pitr" "itre", etc... (Jalam & Teytaud, 2001). Des études ont montré que des systèmes de recherches d'information à base des n-grammes ont préservé leurs performances malgré une déformation de 30%, contrairement à un système à base de mots qui commence à se dégrader à partir d'un taux de 10% de déformations (Miller & all, 1999).

6.3.2- Pondération des termes

Plusieurs options s'offrent à ce stade du processus, certainement la plus utilisée c'est TF-IDF, mais puisque notre objectif c'est chercher plus d'efficacité avec des résultats acceptables, et ne pas alourdir l'algorithme avec des calculs supplémentaires, nous avons choisi d'utiliser, dans l'ensemble de nos expérimentations, la façon la plus simple pour calculer cette pondération à savoir la fréquence du terme dans le document ou dans la catégorie.

6.3.3- Naïve Bayes

Naïve Bayes classifier est le représentant le plus populaire des classifieurs probabilistes et son théorème est au cœur de la problématique de la classification. C'est l'une des méthodes les plus pratiques d'apprentissage, avec les kPPV, Rocchio, les SVM, les arbres de décision, les réseaux de neurones. En revanche si les modèles vectoriels fonctionnent bien, leur fondement est entièrement empirique. Ils sont le résultat de nombreuses années de test. Le modèle probabiliste, au contraire, s'appuie sur une base théorique précise.

Le classifieur bayésien naïf reste un des outils de catégorisation de documents les plus pratiques en raison de ses performances reconnues dans ce domaine, et est aujourd'hui intégré à de nombreux produits commerciaux. Il s'appuie sur un modèle de génération d'un document à partir duquel on peut déduire la ou les classes les plus probables d'appartenance du document. Les paramètres du modèle sont estimés à partir d'un corpus d'apprentissage. Plusieurs expériences ont démontré les bonnes performances de l'algorithme. L'équipe Microsoft a développé son propre algorithme NB connu par MNB (**M**icrosoft **N**aïve **B**ayes) qui est un algorithme de classification fourni par Microsoft **S**QL **S**erver 2005 **A**nalysis **S**ervices (SSAS), conçu pour la modélisation prédictive. (Présenté dans l'annexe MNB).
Une autre application réussie à base du classifieur NB, celle utilisée pour enseigner l'âne Ditto les bases de la langue anglaise. (Présentée dans l'annexe Ditto-The donkey)

Historiquement plusieurs travaux de classification à base de l'approche Naive Bayes ont été développés, citons :

Maron (1961) – Indexation automatique
Mosteller and Wallace (1964) – Identification des auteurs
Van Rijsbergen, Robertson, Sparck Jones, Croft, Harper (1970) – moteurs de recherche
Sahami, Dumais, Heckerman, Horvitz (1998) – Filtres anti-spams.
Adventure Works Cycle (2009) – Marketing (Ciblage de clientèle)

Le principe qui régit et donne son nom à l'algorithme est très simple. Il indique simplement que les différents attributs (dans le cas du texte, les différents termes présents dans le document) sont considérés comme indépendants. C'est la même hypothèse que pour le modèle vectoriel mais cette fois exprimée explicitement dans le cadre de la théorie probabiliste.

Avant de justifier les raisons qui ont motivé l'adoption de cette méthode dans notre approche proposée, nous avons préférer rappeler quelques définitions nécessaires pour une bonne maîtrise du classifieur Naïve Bayes.

Notons que nous avons repris, dans cette section, quelques définitions proposées dans (www.fr.wikipedia.org)

6.3.3.1- Probabilité conditionnelle

La notion de probabilité conditionnelle permet de tenir compte dans une estimation d'une information complémentaire. Par exemple, si je tire au hasard une carte d'un jeu, j'estime naturellement à une chance sur quatre la probabilité d'obtenir un cœur ; mais si j'aperçois un reflet rouge sur la table, je corrige mon estimation à une chance sur deux. Cette seconde estimation correspond à la probabilité d'obtenir un cœur sachant que la carte est rouge. Elle est conditionnée par la couleur de la carte ; donc, conditionnelle.
En théorie des probabilités, la probabilité conditionnelle d'un événement ***A***, sachant qu'un autre événement **B** de probabilité non nulle s'est réalisé est noté $P(A/B)$ défini par :
Le réel $\mathbb{P}(A/B)$ se lit « probabilité de ***A***, sachant ***B*** ». $\mathbb{P}(A/B)$ se note aussi parfois $\mathbb{P}_B(A)$
Mathématiquement, soient (Ω , $\mathcal{B}$, $\mathbb{P}$), un espace probabilisé et ***B*** un événement de probabilité non nulle. À tout événement ***A*** de $\mathcal{B}$, nous associons le nombre noté $\mathbb{P}(A/B)$ ou $\mathbb{P}_B(A)$ défini par:

$$\mathbb{P}_B(A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

Nous pourrions vérifier que l'application $\mathbb{P}_B$ définie par $A \rightarrow \mathbb{P}_B(A)$ est une probabilité.

Et $P(A/B)+P(\neg A/B)=1$

6.3.3.2- Théorème de Bayes

Le théorème de Bayes ou le modèle d'indépendance conditionnelle (Naïve Bayes classifier), qui a été proposé par le mathématicien anglais Thomas Bayes (1702-1761), fournit un cadre théorique pour la problématique de la classification. Dans cette approche, tous les paramètres, sont considérés comme des variables aléatoires issues d'une distribution de probabilité.

Le théorème de Bayes est utilisé dans l'inférence statistique pour mettre à jour ou réviser les estimations d'une probabilité ou d'un paramètre quelconque, à partir des observations et des

lois de probabilité de ces observations. Il y a une version discrète et une version continue du théorème.
En théorie des probabilités, le théorème de Bayes énonce des probabilités conditionnelles : étant donné deux évènements ***A*** et ***B***, le théorème de Bayes permet de déterminer la probabilité de ***A*** *sachant* ***B***, si l'on connaît les probabilités :

- de ***A*** ;
- de ***B*** ;
- de ***B*** *sachant* ***A***.

Ce théorème élémentaire (originellement nommé « de probabilité des causes ») a des applications considérables.
Pour aboutir au théorème de Bayes, on part d'une des définitions de la probabilité conditionnelle :

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$$

en notant $P(A \cap B)$ la probabilité que ***A*** et ***B*** aient tous les deux lieu. En divisant de part et d'autre par *P*(***B***), on obtient :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Qui donne bien le théorème de Bayes.

Chaque probabilité du théorème de Bayes a une dénomination usuelle.

- $\mathbb{P}$ (***A***) est la *probabilité a priori* de ***A,*** elle est « antérieure » au sens qu'elle précède toute information sur ***B***, *P*(***A***) est aussi appelée la *probabilité marginale* de ***A***.
- De même, $\mathbb{P}$ (***B***) est appelé la *probabilité marginale* ou *a priori* des donnés d'apprentissage ***B***.
- $\mathbb{P}$ (***A/B***) est appelée la *probabilité a posteriori* de ***A*** sachant ***B*** (ou encore de ***A*** sous condition ***B***), elle est « postérieure », au sens qu'elle dépend directement de ***B***.
- $\mathbb{P}$ (***B/A***), pour un ***B*** connu, est appelé la *fonction de vraisemblance* de ***A*** (ou encore de ***B*** sous condition ***A***)

6.3.3.3- Inférence bayésienne

On nomme inférence bayésienne la démarche logique permettant de calculer ou actualiser la probabilité d'une hypothèse. Le raisonnement bayésien est appliqué à la prise de décision, on utilise la connaissance des événements pour prédire des événements futurs. Cette démarche est régie par l'utilisation de règles strictes de combinaison des probabilités, desquelles dérive le théorème de Bayes.

Exemple d'inférence bayésienne :

D'où vient ce biscuit ?

(Cet exemple est extrait de l'article anglophone www.wikipedia.en)

Imaginons deux boîtes de biscuits.

L'une, ***A***, comporte 30 biscuits au chocolat et 10 ordinaires.
L'autre, ***B***, en comporte 20 de chaque sorte.
On choisit les yeux fermés une boîte au hasard, puis dans cette boîte un biscuit au hasard. Il se trouve être au chocolat. De quelle boîte a-t-il le plus de chances d'être issu, et avec quelle

probabilité ? Intuitivement, on se doute que la boîte ***A*** a plus de chances d'être la bonne, mais de combien ?

Notons H_A la proposition « le gâteau vient de la boîte ***A*** » et H_B la proposition « le gâteau vient de la boîte ***B*** ». Dans un contexte de classification ***A*** et ***B*** c'est des classes et D est un document.

Si lorsqu'on a les yeux bandés les boîtes ne se distinguent que par leur nom, nous avons $\mathbb{P}(H_A) = \mathbb{P}(H_B)$, et la somme fait 1, puisque nous avons bien choisi une boîte, soit une probabilité de 0,5 pour chaque proposition.

Notons ***D*** l'événement désigné par la phrase « le gâteau est au chocolat ». Connaissant le contenu des boîtes, nous savons que :

- $\mathbb{P}(D / H_A) = 30/40 = 0{,}75$
- $\mathbb{P}(D / H_B) = 20/40 = 0{,}5$

La formule de Bayes nous donne donc :

$$P(H_A|D) = \frac{P(H_A)\cdot P(D|H_A)}{P(H_A)\cdot P(D|H_A)+P(H_B)\cdot P(D|H_B)}$$

$$= \frac{0{,}5\times 0{,}75}{0{,}5\times 0{,}75+0{,}5\times 0{,}5}$$

$$= 0{,}6$$

$\mathbb{P}(H_A / D)$ représente la probabilité d'avoir choisi la boîte ***A*** sachant que le gâteau est au chocolat.

Avant de regarder le gâteau, notre probabilité d'avoir choisi la boîte ***A*** était $\mathbb{P}(H_A)$, soit 0,5.

Après l'avoir regardé, nous révisons cette probabilité à $\mathbb{P}(H_A /D)$, qui sera 0,6. L'observation ***D*** « le gâteau est au chocolat » nous a donc apporté une augmentation de 10% sur la probabilité initiale de H_A « le gâteau vient de la boîte ***A*** ».

Et puisque $\mathbb{P}(H_A /D) + \mathbb{P}(H_B /D) = 1$ (pas d'autre possibilité que d'avoir choisi la boîte ***A*** ou la boîte ***B*** sachant que le gâteau est au chocolat), la probabilité d'avoir choisi la boîte ***B*** sachant que le gâteau est au chocolat est donc de 1 - 0,6 = 0,4.

6.3.3.4- La classification naïve bayesienne

La classification naïve bayesienne est un type de classification bayesienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses.

Définition de l'indépendance :

« Two events ***A*** and ***B*** are statistically independent if the probability of ***A*** is the same value when ***B*** occurs, when ***B*** does not occur or when nothing is known about the occurrence of ***B*** »

«Deux événements ***A*** et ***B*** sont statistiquement indépendants si la probabilité de ***A*** est la même lorsque ***B*** se réalise, ou lorsque ***B*** ne se réalise pas, ou encore quand on ne sait rien sur ***B*** »

A et ***B*** sont indépendants alors : $\mathbb{P}(A|B) = \mathbb{P}(A)$

Exemple :

Supposons qu'il ya deux événements:

A : Amine enseigne la classe sinon c'est Abderahim
B : Il pleut

Nous remarquons que la météo ne dépend pas et n'as pas d'influence sur lequel des professeurs qui va enseigner la classe.

Cela peut être spécifié très simplement par :

$$\mathbb{P}(A|B) = \mathbb{P}(A)$$

Cette propriété implique les règles suivantes :

- $\mathbb{P}(\neg A / B) = \mathbb{P}(\neg A)$
- $\mathbb{P}(B / A) = \mathbb{P}(B)$
- $\mathbb{P}(B \cap A) = \mathbb{P}(B)\ \mathbb{P}(A)$
- $\mathbb{P}(\neg B \cap A) = \mathbb{P}(\neg B)\ \mathbb{P}(A)$
- $\mathbb{P}(B \cap \neg A) = \mathbb{P}(B)\ \mathbb{P}(\neg A)$
- $\mathbb{P}(\neg B \cap \neg A) = \mathbb{P}(\neg B)\ \mathbb{P}(\neg A)$

6.3.3.5- Maximum A Posteriori (MAP) et Maximum de vraisemblance (ML)

En général, nous cherchons l'hypothèse la plus probable compte tenu des données d'apprentissage :

Maximum A Posteriori (MAP)

- $\mathbf{h}_{MAP} = \arg\max \mathbb{P}(h/d)$ où h appartient à H l'espace des hypothèses et d à l'espace des événements
- $\mathbf{h}_{MAP} = \arg\max \dfrac{P(d/h)\ P(h)}{P(d)}$

P (d) peut être ignoré car il est le même pour toutes les probabilités

- $\mathbf{h}_{MAP} = \arg\max \mathbb{P}(d/h)\ \mathbb{P}(h)$

Exemple :

Pour une classification bi-classe comme par exemple les filtres anti-spam :
L'espace des hypothèses correspond à l'appartenance aux classes, on notera :

$P(c_i/d_i) = \arg\max \mathbb{P}(d_i/c_i)\ \mathbb{P}(c_i)$
$\mathbf{d}_i$ est un document
$\mathbf{c}_1$ correspond à la classe spam
$\mathbf{c}_2$ correspond à la classe non spam
$P(c_i/d_i) = \arg\max\{ P(d_i/c_1) P(c_1) , P(d_i/c_2) P(c_2)\}$

Maximum de vraisemblance (ML)

Si on suppose que les probabilités à priori sont les mêmes pour toutes les classes (exemple des deux boîtes de biscuits) alors on aura :

$P(c_i) = P(c_j)$

Une simplification plus poussée nous conduit à :

- $\mathbf{h}_{ML} = \arg\max P(d_i/c_i)$

L'estimation des paramètres pour les modèles de Bayes naïf utilise la méthode du maximum de vraisemblance.

Le calcul de $P(d_i/c_j)$ dépend du modèle de génération des exemples. Dans le domaine de classification de textes, les plus populaires sont le modèle multivarié de Bernoulli et le modèle multinomial.

6.3.3.6- Le modèle multivarié de Bernoulli

Dans le modèle le plus simple, un document est un vecteur binaire de la taille du vocabulaire. Il est généré par tirage aléatoire : chaque terme du vocabulaire peut être présent ou absent avec une certaine probabilité. Seule la présence/absence des termes est utilisée. Leur nombre d'occurrences dans le document n'a pas d'incidence. Le document se représente comme le résultat du tirage de T variables aléatoires indépendantes : $t_1,\ldots,t_v$. Pour rendre les formules exploitables, on fait de plus l'hypothèse d'indépendance des termes (hypothèse naïve Bayes). $P(d_i/c_j)$ se simplifie alors en un produit de probabilités d'occurrences de chaque terme :

$$P(d_i|c_j) = \prod_{t_k \in d_i} P(t_k|c_j)$$

$$= P(t_1/c_j) * P(t_2/c_j) * \ldots\ldots * P(t_n/c_j)$$

Il est possible d'estimer $P(t_k/c_j)$ à partir des exemples du corpus d'apprentissage. On utilise généralement l'estimateur du maximum de vraisemblance.

Que faire si on retrouve une probabilité nulle $P(t_j / c_i) = 0$? (Un terme absent de tous les documents d'une classe)

Comme les probabilités de chaque terme sont multipliées, il suffit en effet qu'un seul terme dans un document à classer ne soit présent dans aucun document d'une classe (ce qui arrive souvent dans le domaine du texte où beaucoup de termes apparaissent très rarement) pour que la probabilité que le document appartienne à cette classe soit nulle.
La correction Laplacienne ou le lissage de laplace (laplace smoothing) intervient pour éviter ces probabilités nulles. Le lissage effectué pendant l'estimation, qui consiste à ajouter 1 à chaque terme, est indispensable.
Puisque le nombre de termes de la base d'apprentissage est important, l'ajout de 1 sera négligeable. On parvient alors, dans le cas du lissage de laplace, à :

$$P(t_k|c_j) = \frac{1 + \sum_{d_i \in c_j} 1_{ik}}{2 + |c_j|}$$

Avec 1_{ik} =1 si t_k appartient à d_i ,
= 0 sinon.
Pour illustrer le lissage de Laplace : dans une base composée de 750 termes, la probabilité d'un terme absent sera (1+0) / (2+1350) = 0,0007396 au lieu de 0 / 1352 = 0

6.3.3.7- Le modèle multinomial

Dans le modèle précédent, il n'est pas possible d'utiliser les fréquences des termes dans les documents. Pour prendre en compte cette information supplémentaire, un autre modèle plus

complexe a été proposé, et qui a été adopté dans notre approche proposée. Un document est une séquence de mots, chacun étant tiré aléatoirement parmi l'ensemble des mots du vocabulaire. Un document est donc généré par une distribution multinomiale des mots avec autant de tirages que de mots dans le document. Il est ainsi possible de prendre en compte la longueur des documents bien que cela soit rarement fait en pratique. L'hypothèse d'indépendance des termes reste nécessaire.

$$P(d_i|c_j) = \prod_{t_k \in d_i} P(t_k|c_j)$$
$$= P(t_1/c_j) * P(t_2/c_j) *\ldots\ldots* P(t_n/ c_j)$$

Les probabilités sont une nouvelle fois estimées à partir des occurrences des exemples du corpus avec l'estimateur du maximum de vraisemblance avec un lissage de laplace :

$$P(t|c) = \frac{1 + T_{ct}}{\sum_{t' \in V}(1 + T_{ct'})}$$

Ou V c'est le vocabulaire du corpus,
T_{ct} le nombre d'occurrences du terme t dans tous les documents de la classe c,
$T_{ct'}$ le nombre d'occurrences de tous les termes y compris t des documents de la classe c.
Quant à la classification, l'estimation $P(d_i/c_j)$ est calculée suivant le modèle utilisé en remplaçant les probabilités par leur estimateur et ***la classe c_j ayant la probabilité la plus élevée sera attribuée au document d_i.***

6.3.3.8- Description de l'algorithme

Il y a plusieurs variantes de l'algorithme NB, voici une description d'une d'entre elles :

1- Préparer les associations (texte, classe d'appartenance) pour tous les documents d'apprentissage.
2- Préparer le document à classifier sous la forme (termes, nombre d'occurrences).
3- Compter pour chaque classe le nombre de documents appartenant a cette classe.
4- Calculer pour chaque classe la probabilité à priori.
5- Calculer de nombre de termes contenus dans une classe et le nombre d'occurrences de chaque terme dans toutes les classes, ce traitement à effectuer sur tous les termes des documents de training.
6- Calculer pour chaque classe la probabilité à posteriori.
7- La classe ayant la probabilité la plus élevée sera attribuée au document.

6.3.3.8- Avantages de la méthode adoptée (Naïve Bayes Classifier)

Cet algorithme dont le modèle d'apprentissage est très général est utilisé dans de nombreux autres domaines que le texte.
David.D Lewis dans (Lewis, 2004) et Hassane Hilali dans (Hilali, 2009) listent un ensemble d'avantages du classifieur bayésien naïf, parmi lesquelles :

- Algorithme facile et simple à implémenter
- Basée sur une théorie mathématique précise
- Efficacité et rapidité dans l'apprentissage et la classification
- Facile à mettre à jour avec de nouveaux exemples d'apprentissage
- Equivalent à un classifieur linéaire, dans sa rapidité d'application

- L'hypothèse d'indépendance des paramètres assouplit l'algorithme pour qu'il soit favorable pour différents types de données
- Très efficace avec des petits corpus d'apprentissage
- Résiste au bruit existant dans les données d'entrée
- Utile pour la classification déterministe comme pour le Ranking puisque il ordonne les classes par degré d'appartenance pour un texte donné
- Requiert une petite quantité de données d'apprentissage pour estimer les paramètres
- Enfin, le plus important c'est que les méthodes Naïve Bayes donnent de bons résultats

En revanche, l'inconvénient principal à notre avis, c'est bien l'hypothèse d'indépendance entre les descripteurs qui est loin d'être réaliste, mais nous pensons qu'elle n'est pas un handicap majeur dans un contexte de classification.

Tous les avantages cités auparavant et particulièrement la simplicité des calculs, l'efficacité des résultats et la facilité de l'implémentation de cette méthode, au contraire à d'autres techniques plus sophistiquées gourmandes en ressources (gestion de mémoire vive) et en temps d'exécution avec des taux d'amélioration des résultats très minimes, ont stimulé et justifier le choix du modèle d'indépendance conditionnelle (Naïve Bayes classifier) pour nos travaux.

6.3.4- Mesures de performances utilisées pour l'évaluation

Puisque nous disposons de plusieurs modèles de catégorisation, nous devons mesurer la qualité des réponses données par le classifieur.
En principe, choisir *le Rappel* ***(R),****la Précision* ***(P)*** et *la F-measure* ($\boldsymbol{F_1}$) pour évaluer et comparer les différents modèles de catégorisation construits n'a pas besoin d'être justifié, et ce choix est presque automatique puisque ces trois indicateurs sont utilisés régulièrement dans le domaine de classification de textes et la recherche d'information avec succès depuis une trentaine d'années.
$\boldsymbol{F_1}$ permet de combiner, les deux mesures classiques *le Rappel (R)* et *la Précision (P*) pour obtenir une moyenne harmonique entre ces deux indicateurs, définit par :

$$F_1 = \frac{2 * P * R}{P + R}$$

Nous tenons aussi à rappeler que pour une catégorie C_i, *la précision* évalue la qualité du classifieur à ne pas introduire de documents d'une autre catégorie dans C_i. Il s'agit du nombre de documents bien classés sur le nombre de documents classés dans C_i.

$$Précision\ (C_i) = \frac{Nombre\ de\ documents\ bien\ classés\ dans\ C_i}{Nombre\ de\ documents\ classés\ dans\ C_i}$$

L*e rappel*, quant à lui, évalue le degré de complétude, c'est-à-dire le nombre de documents bien classés sur le nombre total de documents de la classe C_i.

$$Rappel\ (C_i) = \frac{Nombre\ de\ documents\ bien\ classés\ dans\ C_i}{Nombre\ de\ documents\ de\ la\ classe\ C_i}$$

Notre objectif est converger ces deux grandeurs vers la valeur un (1) pour que $\boldsymbol{F_1}$ convergera aussi vers 1, un objectif difficile à atteindre puisque un rappel intéressant ne peut être acquis qu'au prix d'une faible précision et vice-versa.

Ces mesures précédentes vont servir à évaluer le système par rapport à une seule classe. Pour une évaluation globale du classifieur par rapport à toutes les classes du corpus, nous avons choisit d'utiliser les mesures *micro moyenne* qui correspondent à une moyenne qui pondère les classes par le nombre de documents qu'elles contiennent. La micro-moyenne accorde donc des poids importants aux catégories ayant beaucoup d'exemples.
La micro-moyenne (traduction de micro-averaging) calcule les mesures rappel et précision de façon globale, cela revient à sommer les cases VP et FP de chaque catégorie pour obtenir la table de contingence globale.
La performance globale du classifieur est indiquée en calculant les différentes moyennes (***P, R, F_1***) qui sont calculées à partir des valeurs cumulées.

La précision, le rappel et la F_1 micro-moyenne sont calculés comme suit :

$$P = \frac{VP}{VP + FP} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FP_i)}$$

$$R = \frac{VP}{VP + FN} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FN_i)}$$

$$F_1 = \frac{2 * P * R}{P + R}$$

6.3.5- Les Systèmes Multi-Agents

Contrairement aux différentes approches décrites dans l'état de l'art basé sur un seul point de vue (Intelligence Artificielle classique : le penseur isolé) , et afin d'améliorer les performances du processus de classification basé sur un seul module logiciel on est ainsi naturellement conduit à chercher à donner plus d'autonomie et d'initiative aux différents modules logiciels en optant pour la distribution de la tâche de classification à un Système-Multi-Agents autonome collaboratif (Intelligence Artificielle distribuée : la communauté de penseurs).
Comme les premiers logiciels étaient construits à l'image que l'on se fait du raisonnement humain, les logiciels d'aujourd'hui (logiciel multi-agents) sont bâtis à l'image que l'on se fait du fonctionnement d'une société d'humains : plusieurs composants (appelés agents) réalisent chacun une tâche spécifique, interagissent et communiquent entre eux pour assurer la cohérence, la complétude et la correction d'une activité globale. Comme toute société d'humains, les agents pourront se réorganiser entre eux et adapter leur comportement à l'évolution de l'environnement (concept d'apprentissage). Mais, il va sans dire que tout ce qui se passe dans la tête des agents est entièrement défini par le concepteur.
L'utilisation d'une architecture Multi-Agents adoptant un comportement social de type « *Fourmis* » peut se présenter alors, comme un vrai remède pour améliorer les performances de notre modèle de classification. Sur des corpus de taille faible, la différence avec un système classique (un seul agent) n'est pas vraiment remarquée mais dés qu'on passe au

traitement des corpus de grande taille ou même infinie comme le Web une telle architecture pourrait être une très bonne solution pour notre problème. Peu de travaux ont déjà plus ou moins traité la problématique de classification de textes à base des SMA (Lumer, 1994), (Monmar, 1999).

On est ainsi naturellement conduit à chercher à donner plus d'autonomie et d'initiative aux différents modules logiciels. Le concept de système multi-agents propose un cadre de réponse à ces deux enjeux complémentaires (et à première vue contradictoires) : **autonomie** et **organisation**.

Dans un contexte de catégorisation automatique de textes, l'autonomie des agents est exprimée dans la première phase du processus par l'attribution des documents aux classes d'une façon indépendante des autres agents, toutefois l'organisation du SMA et la collaboration des agents du système entre eux peut être expliquée par la décision finale de classification du document à une classe choisie après un vote majoritaire des agents.

6.4- Base de texte utilisée pour l'évaluation

Afin de pouvoir comparer les performances obtenues par divers algorithmes, il est nécessaire de les tester sur les mêmes corpus.
Le terme « corpus » désignait à l'origine des sources documentaires sous forme de recueil de textes rassemblant exhaustivement tous les documents pour certains champs d'étude. (Benveniste, 2000). Néanmoins cette notion d'exhaustivité n'est pas toujours possible dans tous les domaines puisque les corpus de langue vivante, par exemple, sont ouverts et des mises à jour sont proposées en permanence
Plus récemment, une définition plus vague du terme « corpus électronique » est apparue, il s'agit d'une collection de textes sous un format compréhensible par l'ordinateur.
De nombreux types de corpus ont vu le jour ces dernières années qui s'organisent en différentes typologies. M.Antoniotti, préfère classifier les corpus en fonction des caractéristiques qui les opposent. (Antoniotti, 2002).

Le tableau suivant résume les types de corpus :

Critère d'opposition	Types de corpus
Langues	Corpus monolingues / Corpus multilingues
Taille	Echantillons / Textes entiers
Evolutivité	Corpus fermés / Corpus ouverts
Thèmes traités	Corpus généralistes / Corpus spécialisés
Pré-traitements des textes	Corpus bruts / Corpus préparés

Tableau 6.1 : Types de corpus

Quelques bases de textes sont donc émergées comme « corpus de référence » pour la catégorisation de textes. Ils doivent regrouper un certain nombre de documents qui sont tenus d'être de diverses utilisations. Ainsi une dimension suffisante et la diversité des documents sont deux caractéristiques principales d'un corpus pour qu'il soit qualifié de « référence ».
L'utilisation de ces corpus standards permet ainsi une comparaison plus aisée des performances des différentes techniques de classification.
On trouve principalement des comparaisons sur la base ***Reuters***, qui est une classification de dépêches de presse. Dans le domaine médical, on se réfère également à la base ***Ohsumed***.
Y.yang a mis cet aspect en évidence dans son étude qui synthétise les performances sur le corpus Reuters en évaluant les résultats obtenus de plusieurs algorithmes d'apprentissages sur divers versions de Reuters. (Yang, 1999).

6.4.1- Présentation générale du corpus Reuters

Reuters est un corpus de dépêches en langue anglaise qui a été proposé par l'agence de presse Reuters en 1987. Il correspond à une problématique de classification en plusieurs classes (un document appartient à une ou plusieurs classes).
Deux qualités principales caractérisent les documents de Reuters c'est qu'ils sont courts et plutôt homogènes avec un vocabulaire riche (environ 17000 mots), et la disponibilité gratuitement de la base dans le Web (http://www.research.att.com/~lewis/reuters21578.html), pour la version Lewis et sur (http://www-2.cs.cmu.edu/~yiming/) pour les versions Yang, Apte et PARC.
Les diverses expérimentations des chercheurs sur la base ont fait de ce corpus comme corpus de référence dans le domaine de la classification supervisée de textes.
Citons à titre d'exemple les auteurs : (Yang & Liu, 1999] avec 13 algorithmes (SVM, RN,AD,NB, etc..), (Schapire & all, 1998) avec Rocchio, (Joachims, 1998) et (Dumais & all, 1998) et (Jalam, 2003) avec (SVM), qui ont utilisé Reuters comme corpus pour apprendre, tester et évaluer les performances de leurs classifieurs.

Le tableau 6.2 illustre un exemple de texte du corpus :

```
<TITLE>AMERICUS TRUST &lt;HPU> EXTENDS DEADLINE</TITLE>
<TEXT>
Americus Trust for American Home
Products Shares said it extended its deadline for accepting
tendered shares until November 26, an extension of nine months.
    The trust, which will accept up to 7.5 mln shares of
American Home Products &lt;AHP>, said it has already received
tenders for about four mln shares.
    The trust is managed by Alex. Brown and Sons Inc &lt;ABSB> and
was formed November 26, 1986.
 </TEXT>
```

Tableau 6.2 : Exemple de texte du corpus Reuters-21578.

(On peut noter l'utilisation du sigle <> pour signaler le nom d'entreprise)

6.4.2- Historique

Ce corpus initialement nommé Reuters-22173 comportait 22173 dépêches de presse qui ont été publiés par en 1987. Tous les articles ont été rassemblés et indexés à des catégories par le personnel de l'agence et le groupe Carnegie (Carnegie Group Inc - CGI).
En 1990, les documents ont été mis à la disposition du laboratoire de recherche d'information (Université de Massachusetts à Amherst), par Reuters et CGI, pour des fins de recherche. Le formatage des documents et la génération de fichiers associés a été réalisée en 1990 par David D. Lewis et Stephen Harding au même laboratoire.
Un autre formatage et génération des fichiers associés a été fait été fait en 1991 et 1992 par David D. Lewis et Peter Shoemaker au Centre d'information et études du langage (Université de Chicago). Cette base de fichiers a donné naissance au 01/01/1993, à la première version dénommée "Reuters-22173"

6.4.3- Evolution du corpus

Lors de la conférence ACM SIGIR 96 en août 1996, un groupe de chercheurs dans le domaine de catégorisation de texte ont débattu une démarche qui pourrait faire de Reuters-22173 corpus de référence dans toutes les études. Il a été a décidé qu'une nouvelle version de la base

doit être produite avec moins d'ambiguïtés, avec une orthographe soignée. L'occasion était également parfaite, pour corriger un certain nombre d'erreurs typographiques, des erreurs dans la catégorisation et dans le formatage des documents. Steve Finch et David D. Lewis ont réalisé ce travail de Septembre à Novembre 1996, se basant essentiellement sur le SGML. Le résultat était la suppression de 595 documents qui étaient des répliques exactes d'autres documents du corpus. Le nouveau corpus épuré contenait donc 21 578 documents, et est donc appelé Reuters-21578 collection.
Initialement, Reuters-21578 est disponible dans 22 fichiers. Chacun des 21 premiers fichiers (reut2-000.sgm jusqu'à reut2-020.sgm) contient 1000 documents, tandis que le dernier (reut2-021.sgm) contient 578 documents. Les fichiers sont sous format SGML.
Une autre mise à jour faite par Lewis en écartant 1765 textes sans catégories prédefinies, qui sont inutiles dans un contexte d'apprentissage supervisé. Cette version est connue par Reuters "ModLewis" composée de 13265 documents pour l'ensemble d'apprentissage, et 6188 documents pour l'ensemble de test.
Depuis, plusieurs versions ont été diffusées, les différences entre ces versions concernent essentiellement le nombre des catégories du corpus, ainsi que la manière de définir le découpage des corpus d'apprentissage et de test.
Ainsi pour évaluer les méthodes de catégorisation, la collection de textes Reuters est généralement répartie en deux ensembles : l'ensemble d'apprentissage (textes pré-catégorisés) et l'ensemble de test (textes à catégoriser).
Le découpage le plus souvent rencontré se nomme découpage *Apté* du nom des premiers auteurs à l'avoir proposé (Apté & all, 1994). La base d'apprentissage initiale est constituée des documents antérieurs au 8 avril 1987, soit 9603 documents, et la base de test de tous les documents ultérieurs, soit 3299 documents, soit 8676 documents écartés.
Malheureusement, il existe de légères modifications à ce découpage qui rendent certaines comparaisons difficiles. Ainsi (Yang & Liu, 1999) ont supprimé de la base de test tous les documents qui n'appartiennent à aucune catégorie : ils n'utilisent que 3019 documents sur la base de test. La suppression de ces documents ne peut qu'améliorer les résultats par rapport au découpage traditionnel, puisque les risques de mauvais classement sont réduits. (Dumais & all, 1998) considèrent 118 catégories : certaines catégories n'ont donc pas de documents étiquetés sur la base de test, et la façon dont ces catégories sont prises en considération dans leur évaluation n'est pas évidente.

Le tableau 6.3 montre 5 versions proposées parmi d'autres, avec les statistiques concernant chacune d'elles :

Nombre de catégories	Nombre des documents d'apprentissage	Nombre des documents de test	Nombre global des documents	Corpus
182	21450	723	22173	Reuters- 22173
182	20856	722	21578	Reuters-21578a
135	14704	6746	21578	Reuters-21578b
113	13625	6188	19813	Reuters-ModLewis
118	9603	3299	12902	Reuters-ApteMod94
90	9586	3745	13331	Reuters-ApteModb
10	7194	2788	9982	Reuters-Top10

Tableau 6.3 : Principales versions de la collection Reuters

En conséquent, il reste délicat de comparer les performances obtenues sur les différentes versions du corpus. Cependant, les caractéristiques globales du corpus sont restées identiques, et les remarques sur le comportement général des systèmes étudiés sont toujours valables.
N'empêche que les réelles comparaisons restent l'évaluation des différents classifieurs sur les mêmes versions du corpus. Comme c'est le cas dans nos expérimentations qui vont être effectuées sur la même version du corpus à savoir ***Reuters21578-Top10*** qui sera tout simplement nommé corpus **Reuters** dans la suite de ce mémoire.

6.4.4- Définition des catégories du corpus Reuters-21578-ApteMod

Les mises à jour Reuters-21578 ApteMod ou ModeApté, ont été obtenues par la suppression des documents non étiquetés que comportait la version précédente (textes ambigus) , elles ont permis aussi de supprimer les documents présents deux fois, de corriger des erreurs typographiques, et d'autre part par la conservation des catégories ayant au moins un document dans la base d'apprentissage et un dans la base de test. Pour se ramener à moins d'une centaine de catégories. Il en résulte 90 catégories avec 9586 documents pour l'ensemble d'apprentissage et 3745 pour l'ensemble de test.

Les 90 catégories issues du découpage des ensembles d'apprentissages et de tests sont présentées dans le tableau 6.4, ainsi que le nombre de documents associés disponibles sur chaque base. Elles sont classées par ordre décroissant du nombre de documents associés sur la base d'apprentissage. Le nombre de documents disponibles pour effectuer l'apprentissage décroît rapidement ; dès la vingt-sixième catégorie, ce nombre est inférieur à cinquante. Il faut noter que les documents sont à peu près également répartis sur les deux bases, c'est-à-dire que les catégories ayant beaucoup (respectivement peu) de documents associés à la base d'apprentissage ont également beaucoup (respectivement peu) de documents associés à la base de test.

	Catégorie	Apprentissage	Test
1	Earn	2877	1087
2	Acquisition	1650	719
3	Money-fx	538	179
4	Grain	433	149
5	Crude	389	189
6	Trade	369	118
7	Interest	347	131
8	Wheat	212	71
9	Ship	197	89
10	Corn	182	56
11	Money-supply	140	34
12	Dlr	131	44
13	Sugar	126	36
14	Oilseed	124	47
15	Coffee	111	28
16	Gnp	101	35
17	Gold	94	30
18	Veg-oil	87	37
19	Soybean	78	33
20	Nat-gas	75	30
21	Bop	75	30
22	Livestock	75	24

	Catégorie	Apprentissage	Test
46	Tin	18	12
47	Rapeseed	18	9
48	Orange	16	11
49	Housing	16	4
50	Strategic-metal	16	11
51	Hog	16	6
52	Lead	15	14
53	Soy-oil	14	11
54	Heat	14	5
55	Soy-meal	13	13
56	Fuel	13	10
57	Lei	12	3
58	Sunseed	11	5
59	Dmk	10	4
60	Lumber	10	6
61	Tea	9	4
62	Income	9	7
63	Oat	8	6
64	Nickel	8	1
65	L-cattle	6	2
66	Groundnut	5	4
67	Instal-debt	5	1

23	Cpi	69	28
24	Reserves	55	18
25	Cocoa	55	18
26	Carcass	50	18
27	Copper	47	18
28	Jobs	46	21
29	Yen	45	14
30	Ipi	41	12
31	Iron-steel	40	14
32	Cotton	39	20
33	Gas	37	17
34	Barley	37	14
35	Rubber	37	12
36	Alum	35	23
37	Rice	35	24
38	Palm-oil	30	10
39	Meal-feed	30	19
40	Sorghum	24	10
41	Retail	23	2
42	Zinc	21	13
43	Silver	21	8
44	Pet-chem	20	12
45	Wpi	19	10

68	Rape-oil	5	3
69	Platinum	5	7
70	Sun-oil	5	2
71	Jet	4	1
72	Coconut	4	2
73	Coconut-oil	4	3
74	Potato	3	3
75	Propane	3	3
76	Cpu	3	1
77	Copra-cake	2	1
78	Palmkernel	2	1
79	Naphtha	2	4
80	Palladium	2	1
81	Rand	2	1
82	Dfl	2	1
83	Nzdlr	2	2
84	Rye	1	1
85	Cotton-oil	1	2
86	Lin-oil	1	1
87	Castor-oil	1	1
88	Sun-meal	1	1
88	Groundnut-oil	1	1
90	Nkr	1	2

Tableau 6.4 : Répartition des documents par catégorie, avec le nombre de documents associés pour l'apprentissage et le test

Ce corpus souffre d'une mauvaise définition de ces catégories, et d'un grand déséquilibre dans la répartition des documents entre ces catégories. En effet, il existe des catégories qui sont favorisées par rapport aux autres en termes de nombre des documents présents dans les jeux de tests et apprentissage. La figure 6.1 montre que seules les 20 premières catégories contiennent plus de 100 textes.

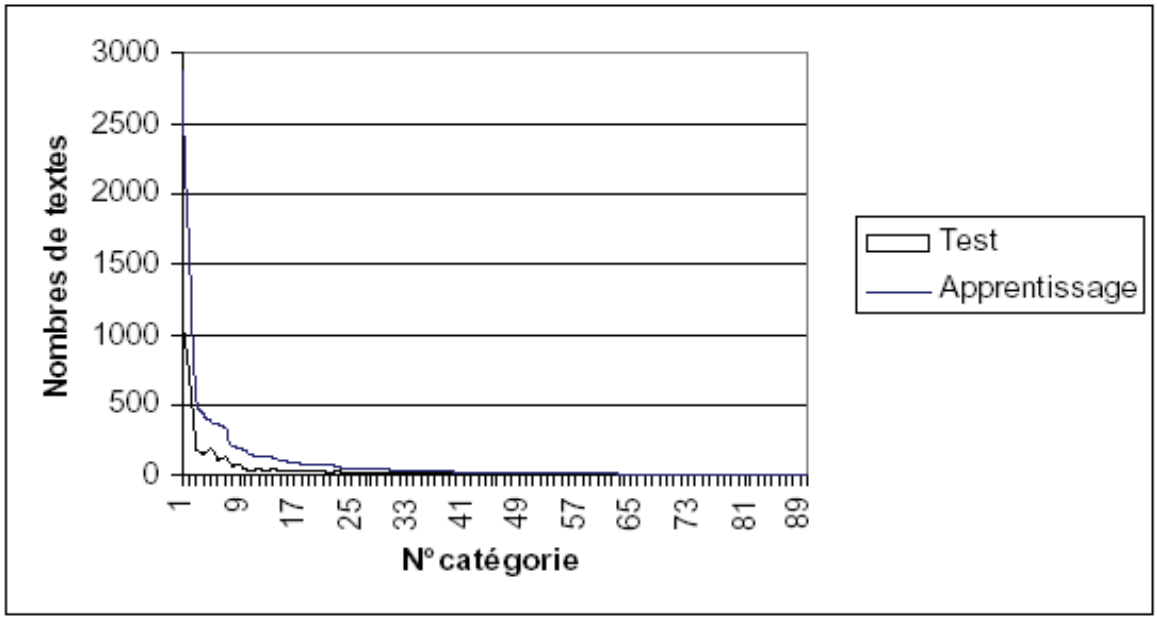

Figure 6.1 : Nombre de textes par catégories de la collection Reuters - (Jalam, 2003)

6.4.5- Reuters21578-ModeApté[10]

Plusieurs expériences ont été réalisées sur la base de textes Reuters-21578. Pour nos travaux, l'ensemble de nos évaluations a été réalisé sur une sous-collection de cette base connue par Reuters21578-Top10 que nous avons utilisé pour entraîner et tester notre classifieur.
Puisque notre objectif est d'associer un texte à une catégorie exclusive, nous avons opté pour une version du corpus qui ne contient ni les textes non étiquetés, ni les textes de multiples étiquettes. En outre, toutes les classes mal représentées avec moins de 150 dépêches d'apprentissage et moins de 50 dépêches de test ont été éliminés. L'ensemble des dépêches qui en résulte est 9982 dont 7194 textes d'apprentissage et 2788 de test. Les 9982 documents, correspondent aux 10 classes les plus représentées dans le corpus. Le tableau 6.5 illustre la répartition de ces 9 982 documents sur les 10 classes :

	Catégorie	Apprentissage	Test	Total catégorie
1	Earn	2877	1087	3964
2	Acquisition	1650	719	2369
3	Money-fx	538	179	717
4	Grain	433	149	582
5	Crude	389	189	578
6	Trade	369	118	487
7	Interest	347	131	478
8	Wheat	212	71	283
9	Ship	197	89	286
10	Corn	182	56	238
Total Général		**7194**	**2788**	**9982**

Tableau 6.5 : Reuters21578-Top10

Reuters-Top10 est donc, une version abrégée du corpus Reuters-21578, qui conserve seulement les 10 premières catégories ayant le plus d'effectifs en nombre de documents associés, toutefois elle compte presque 50% des documents du corpus Reuters21578 1^{ére} version et presque 80% de la version "ModApté".
Certainement, manipuler 80% du corpus avec 10 catégories seulement assouplit le traitement et allège considérablement le processus. Donc de toute évidence et d'une manière générale traiter, comparer et présenter les résultats de 10 catégories est plus pratique que 90. Enfin, nous tenons à préciser que nous ne sommes pas les seuls à choisir et opter pour le Top10 de Reuters mais plusieurs auteurs ont confirmé dans leurs travaux qu'un classifieur qui réussit à classer dans les 10 catégories les plus représentées de Reuters, ne va pas échouer dans les autres. En revanche le seul inconvénient qui s'oppose c'est bien les possibilités de comparaisons qui se rétrécirent seulement aux expériences qui ont testé leurs classifieurs avec Reuters-Top10.

6.5- Applications opérationnelles

Dans cette partie on expose comment les techniques décrites dans ce mémoire ont été intégrées dans une application opérationnelle de catégorisation des dépêches de Reuters.
Dans une première application, ces modèles sont utilisés pour classer les différents documents du corpus avec une approche classique non distribuée basée sur une seule entité logicielle avec une variation dans la représentation des documents (2, 3, 4, 5, 6, et 7-grammes).

Contrairement à la première application, la deuxième application est basée sur une approche distribuée dans laquelle on va déléguer la tâche de classification à un Système Multi-Agents (SMA) autonome collaboratif.
Nous présentons dans cette section une description de l'ensemble de nos expérimentations avec les deux approches distribuées et non distribuées suivie d'une comparaison des différents résultats qui sera développé en fin de cette section.

6.5.1- Environnement de développement

Java est le nom d'une technologie mise au point par Sun Microsystems qui permet de produire des logiciels indépendants de toute architecture matérielle. Cette technologie s'appuie sur différents éléments qui, par abus de langage, sont souvent tous appelés Java :

- le **langage Java** est un langage de programmation orienté objet ;
- un programme java s'exécute dans une machine virtuelle, dite machine virtuelle Java ;
- le bytecode Java est le résultat de la compilation d'un programme écrit en Java par le compilateur Java ;
- la plate-forme Java correspond à la machine virtuelle Java plus des spécifications d'API :
    - Java Platform, Standard Edition (Java SE) contient les API de base et est destiné aux ordinateurs de bureau ;
    - Java Platform, Enterprise Edition (Java EE) contient, en plus du précédent, les API orientées entreprise et est destiné aux serveurs ;
    - Java Platform, Micro Edition (Java ME) est destiné aux appareils mobiles tels que assistants personnels ou smartphones ;

Le **langage Java** est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

L'utilisation native du langage Java pour des applications sur un poste de travail restait jusqu'à présent relativement rare à cause de leur manque de rapidité. Cependant, avec l'accroissement rapide de la puissance des ordinateurs, les améliorations au cours de la dernière décennie de la machine virtuelle Java et de la qualité des compilateurs, plusieurs technologies ont gagné du terrain comme par exemple Netbeans et l'environnement Eclipse, les technologies de fichiers partagés Limewire et Azureus. Java est aussi utilisé dans le programme de mathématiques Matlab, au niveau de l'interface homme machine et pour le calcul formel. Les applications Swing apparaissent également comme une alternative à la technologie .NET. (www.fr.wikipedia.org)

Ainsi notre choix a été justifié par ces avantages qui s'ajoutent au fait que la plateforme Java avec tous ses éléments est téléchargeable gratuitement.

6.5.2- Approche non distribuée

6.5.2.1- Démarche à suivre

- **Prétraitements**
  - Conversion des majuscules en minuscule, éliminer le signes de ponctuations ()[]{}=:?!;-,_"+*/.",<>≥≤%«»&, de même pour les chiffres qui ne sont pas pris en compte.
  - Segmentation des textes en n-grammes.
  - Construire la liste du vocabulaire de tous les termes distincts qui apparaissent dans tous le corpus d'apprentissage.
  - Calcul des fréquences des termes (attributs) dans les documents d'apprentissage.
  - Tous les documents du jeu d'apprentissage sont transformés en une matrice des fréquences des termes dont les colonnes sont les dix classes du corpus et les lignes correspondent à tous les termes du vocabulaire.
- **Apprentissage**
  - Entrainer le classifieur sur le corpus d'apprentissage, en calculant les probabilités à priori des dix classes et les vraisemblances de tous les termes du vocabulaire relatives à ces classes.
- **Test**
  - Tester le classifieur en utilisant les documents du corpus test, en calculant les probabilités à posteriori d'appartenance des documents test aux différentes classes.
  - Classer les documents dans les classes qui disposent des plus grandes probabilités à posteriori.
  - Générer les matrices de contingence correspondantes : Vrai Positif (**VP**), Faux Positif (**FP**), Faux Négatif (**FN**), Vrai Négatif(**VN**).
  - Calculer les mesures de performances Précision/Rappel/F-measure.
- **Choix du meilleur classifieur**
  - Répéter le même processus pour 2, 3, 4, 5, 6 et 7-grammes.
  - Evaluer et comparer les résultats des différents classifieurs construits, par les mesures de performances calculées précédemment.
  - Le modèle de classification qui fournit les meilleures F-measure ($\boldsymbol{F_1}$) sera adopté pour l'approche distribuée.
- **Classification de nouveaux textes**
  - Conversion des majuscules en minuscule, éliminer les mêmes signes de ponctuations, de même pour les chiffres.
  - Segmentation du texte en 4-grammes.
  - Calculer les probabilités à posteriori d'appartenance du document aux dix classes.
  - Le document est assigné à la catégorie ayant la probabilité à postériori la plus grande.

6.5.2.2- Résultats expérimentaux

- **Matrices de contingence**

  Les documents correctement classés sont des VP ;

Les documents correctement non classés sont des VN ;
Les documents incorrectement classés sont des FP ;
Les documents incorrectement non classés sont des VN.

Catégorie Ci		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**VP_i**	**FP_i**
	Non	**FN_i**	**VN_i**

■ **Algorithme de construction des matrices**

Début
Répéter pour tous les documents du corpus de test :
Si C_i_Doc est correctement attribué à C_i
Alors
$VP_i=VP_i+1$;
$VN_j=VN_j+1$; pour toutes les classes autre que C_i

Sinon C_i_Doc est incorrectement attribué à C_j
$FN_i=FN_i+1$; $FP_j=FP_j+1$
Fin de la boucle Répéter
Ecrire VP, FP, FN, VN des 10 classes
Fin de l'algorithme

■ **Calcul de précision, rappel et F-measure**

Pour chaque classe : $R_i = \frac{VP_i}{VP_i + FN_i}$, $P_i = \frac{VP_i}{VP_i + FP_i}$, $F1_i = \frac{2 * P_i * R_i}{P_i + R_i}$

Les mesures MicroMoyennes :

$$P = \frac{VP}{VP + FP} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FP_i)}, \quad R = \frac{VP}{VP + FN} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FN_i)}, \quad F_1 = \frac{2 * P * R}{P + R}$$

- Dérouler la démarche avec des textes représentés en 2-grammes

R=90.3%
P=97.0%
F_1=93.6%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	982	30
	Non	105	707

R=53.4%
P=90.6%
F_1=67.2

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	384	40
	Non	335	1305

R=57.0%
P=21.5%
F_1=31.2%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	102	372
	Non	77	1587

R=4.0%
P=50.0%
F_1=7.5%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	6	6
	Non	143	1683

R=3.2%
P=7.5%
F_1=6.1%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	6	2
	Non	183	1683

R=18.8%
P=56.4%
F_1=28.2%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	22	17
	Non	95	1667

R=86.3%
P=19.2%
F_1=31.3%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	113	477
	Non	18	1576

P=53.5%
R=35.2%
F_1=42.5%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	38	70
	Non	33	1651

R=16.9%
P=55.6%
F_1=25.9%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	15	12
	Non	74	1674

R=42.1%
P=52.3%
F_1=36.2%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	21	72
	Non	35	1668

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**1689**	**1098**
	Non	**1098**	**15201**

Tableaux 6.6 : Matrices de contingence 2-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 2-grammes est

$$F_1=60.6\%$$

- Répéter la même démarche avec des textes représentés en 3-grammes

R=96.0%
P=91.4%
$F_1=93.6\%$

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1043	98
	Non	44	1117

R=80.9%
P=94.6%
$F_1=87.3\%$

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	582	33
	Non	137	1578

P=71.5%
R=41.2%
$F_1=52.2\%$

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	128	183
	Non	51	2032

R=26.2%
P=53.4%
$F_1=35.1\%$

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	39	34
	Non	110	2121

R=40.2%
P=93.8%
$F_1=56.3\%$

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	76	5
	Non	113	2084

R=76.9%
P=61.2%
$F_1=68.2\%$

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	90	57
	Non	27	2070

R=77.9%
P=55.7%
F_1=65.0%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	102	81
	Non	29	2058

R=63.4%
P=39.5%
F_1=48.6%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	45	69
	Non	26	2115

R=37.1%
P=63.5%
F_1=46.8%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	33	19
	Non	56	2127

R=39.3%
P=31.4%
F_1=34.9%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	22	48
	Non	34	2138

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2160**	**627**
	Non	**627**	**19440**

Tableaux 6.7 : Matrices de contingence 3-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 3-grammes est

F_1=77.5%

- Le même processus avec des textes représentés en 4-grammes

R=93.5%
P=92.0%
F_1=92.7%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1016	88
	Non	71	1229

R=90.5%
P=90.2%
F_1=90.4%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	651	71
	Non	68	1594

R=76.0%
P=74.3%
F_1=75.1%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	136	47
	Non	43	2109

R=40.9%
P=50.4%
F_1=45.2%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	61	60
	Non	88	2184

R=52.4%
P=86.6%
F_1=65.3%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	99	15
	Non	90	2146

R=88.0%
P=46.6%
F_1=60.9%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	103	118
	Non	14	2142

R=71.8%
P=66.2%
F_1=68.9%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	94	48
	Non	37	2151

R=56.3%
P=42.6%
F_1=48.5%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	40	54
	Non	31	2205

R=30.3%
P=64.3%
F_1=41.2%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	27	15
	Non	62	2218

R=32.1%
P=40.9%
F_1=36.0%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	18	26
	Non	38	2227

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2245**	**542**
	Non	**542**	**20205**

Tableaux 6.8 : Matrices de contingence 4-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 4-grammes est

$$F_1=80.6\%$$

- Ensuite avec des textes représentés en 5-grammes

R=80.1%
P=91.5%
F_1=85.4%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	871	81
	Non	216	1232

R=89.0%
P=84.8%
F_1=86.6%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	640	115
	Non	79	1463

R=84.4%
P=65.1%
F_1=73.5%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	151	81
	Non	28	1952

R=66.4%
P=48.1%
F_1=55.8%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	99	107
	Non	50	2004

R=60.3%
P=77.0%
F_1=67.7%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	114	34
	Non	75	1989

R=91.5%
P=37.4%
F_1=53.1%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	179
	Non	10	1996

R=52.7%
P=65.7%
F_1=58.5%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	69	36
	Non	62	2034

R=35.2%
P=44.6%
F_1=39.4%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	25	31
	Non	46	2078

R=%
P=%
F_1=%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	26	19
	Non	63	2077

R=1.8%
P=50.0%
F_1=3.5%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1	1
	Non	55	2102

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2103**	**684**
	Non	**684**	**18927**

Tableaux 6.9 : Matrices de contingence 5-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 5-grammes est

$$F_1=75.5\%$$

- Répéter la démarche une autre fois avec des textes représentés en 6-grammes

R=77.2%
P=89.5%
F_1=82.9%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	839	98
	Non	248	1196

R=87.2%
P=81.9%
F_1=84.4%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	627	139
	Non	92	1408

R=87.2%
P=60.5%
F_1=71.4%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	156	102
	Non	23	1879

R=79.2%
P=48.0%
F_1=59.7%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	118	128
	Non	31	1917

R=54.5%
P=75.2%
F_1=63.2%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	103	34
	Non	86	1932

R=84.6%
P=35.7%
F_1=50.3%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	99	178
	Non	18	1936

R=47.3%
P=59.0%
F_1=52.5%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	62	43
	Non	69	1973

R=1.4%
P=50%
F_1=2.7%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1	1
	Non	71	2035

R=30.3%
P=55.1%
F_1=39.1%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	27	22
	Non	62	2008

R=7.1%
P=33.1%
F_1=11.8%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	4	8
	Non	52	2031

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2035**	**752**
	Non	**752**	**18315**

Tableaux 6.10 : Matrices de contingence 6-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 6-grammes est

F_1=73,0%

- Et enfin calculer les mêmes mesures avec des textes représentés en 7-grammes

R=74.0%
P=86.8%
F_1=79.9%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	804	122
	Non	283	1138

R=86.9%
P=78.7%
F_1=82.6%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	625	169
	Non	94	1317

R=77.1%
P=59.5%
F_1=67.2%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	138	94
	Non	41	1804

R=65.8%
P=47.3%
F_1=55.1%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	98	109
	Non	51	1844

R=45.0%
P=62.0%
F_1=52.1%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	85	52
	Non	104	1857

R=81.2%
P=32.6%
F_1=46.6%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	95	196
	Non	22	1847

R=48.9%
P=53.3%
F_1=51.0%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	64	56
	Non	67	1878

R=1.4%
P=33.3%
F_1=2.6%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1	2
	Non	71	1942

R=30.3%
P=50.9%
F_1=38.0%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	27	26
	Non	62	1915

R=10.7%
P=24.0%
F_1=14.8%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	6	19
	Non	50	1936

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**1942**	**845**
	Non	**845**	**17478**

Tableaux 6.11 : Matrices de contingence 7-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 7-grammes est

$$F_1=70.3\%$$

- **Choix du N (du N-Grammes)**

  Le modèle de classification de textes représentés en 4-grammes qui donne les meilleurs résultats à savoir les meilleures F_1, sera adopté pour l'approche distribuée.

6.5.3- Approche distribuée

6.5.3.1- Démarche à suivre

- **Prétraitements**
  - Conversion des majuscules en minuscule, éliminer le signes de ponctuations ()[]{}=:?!;-,_"+*/.",<>≥≤%«»&, de même pour les chiffres qui ne sont pas pris en compte.
  - Segmentation des textes en 4-grammes.
  - Construire la liste du vocabulaire de tous les termes distincts qui apparaissent dans des sous-ensembles du corpus d'apprentissage, obtenus en partageant la base de textes sur le nombre d'agents.
  - Calcul des fréquences des termes dans les documents d'apprentissage correspondants.
  - Les documents du mini-corpus d'apprentissage sont transformés en une matrice des fréquences des termes dont les colonnes sont les dix classes du corpus et les lignes correspondent à tous les termes du vocabulaire du mini-corpus.
- **Apprentissage**
  - Entrainer le classifieur multi-agents sur les sous-ensembles du corpus d'apprentissage : Chaque agent exercera son apprentissage sur un échantillon du corpus. Les résultats de l'apprentissage sont les probabilités à priori des dix classes et les vraisemblances de tous les termes du vocabulaire relatives à ces classes, pour chaque agent.
- **Test**
  - Tester le classifieur sur tout le corpus test : chaque document va être traité par l'ensemble de tous les agents, chaque agent dans le système fera sa propre classification pour le même texte.

- Le document sera catégorisé par un agent dans la classe qui possède la plus grande probabilité à posteriori.
- La catégorisation finale du texte sera faite dans la classe qui a été nommé par le plus grand nombre d'agents.
- Générer les matrices de contingence correspondantes.
- Calculer les mesures de performances Précision/Rappel/F-measure.

■ **Choix du meilleur classifieur**

- Répéter le même processus en augmentant le nombre d'agents 3, 9, 21, 33, 61, 99 et 181 agents (Le nombre est choisi impair pour éviter une égalité complète dans les votes et une catégorie l'emportera toujours)
- Evaluer et comparer les résultats des différents classifieurs multi-agents construits, par les mesures de performances calculées précédemment ajouté à un facteur très important à savoir le temps d'exécution.
- Le classifieur multi-agents qui procure les résultats les plus stables sera adopté.

■ **Classification de nouveaux textes**

- Conversion des majuscules en minuscule, éliminer les mêmes signes de ponctuations, de même pour les chiffres.
- Segmentation du texte en 4-grammes.
- Chaque agent du SMA va calculer indépendamment les probabilités à posteriori d'appartenance du document aux dix classes.
- Le document est assigné à la catégorie élue par le plus grand nombre d'agents.

6.5.3.2- Résultats expérimentaux

■ **Construction des matrices de contingence pour 3, 9, 21, 33, 61, 99 et 181 agents**

■ **Calcul de précision, rappel et F-measure**

- Exécuter le processus par un SMA composé de 3 agents

R=93.7%
P=92.6%
F_1=93.2%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1019	81
	Non	68	1225

R=91.2%
P=90.6%
F_1=90.9%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	656	68
	Non	63	1594

R=73.7%
P=73.3%
F_1=73.5%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	132	48
	Non	47	2109

R=40.9%
P=51.7%
F_1=45.7%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	61	57
	Non	88	2174

R=54.0%
P=86.4%
F_1=66.4%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	102	16
	Non	87	2136

R=93.2%
P=48.7%
F_1=63.9%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	109	115
	Non	8	2141

R=69.5%
P=66.4%
F_1=67.9%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	91	46
	Non	40	2150

R=57.7%
P=42.7%
F_1=49.1%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	41	55
	Non	30	2200

R=34.8%
P=70.5%
F_1=46.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	31	13
	Non	58	2210

R=37.5%
P=45.7%
F_1=41.2%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	21	25
	Non	35	2225

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2263	524
	Non	524	20164

Tableaux 6.12 : Matrices de contingence 3 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 3 agents est

$$F_1=81.2\%$$

- Répéter le même processus par 9 agents

R=94.1%
P=93.0%
$F_1=93.6\%$

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1023	77
	Non	64	1215

R=92.4%
P=92.0%
$F_1=92.2\%$

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	664	58
	Non	55	1587

R=74.9%
P=73.6%
$F_1=74.2\%$

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	134	48
	Non	45	2105

R=43.0%
P=50.8%
$F_1=46.5\%$

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	64	62
	Non	85	2159

R=56.6%
P=87.0%
$F_1=68.6\%$

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	16
	Non	82	2130

R=95.7%
P=51.1%
$F_1=66.7\%$

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	112	107
	Non	5	2141

R=71.0%
P=69.4%
$F_1=70.2\%$

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	93	41
	Non	38	2136

R=63.4%
P=47.9%
$F_1=54.5\%$

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	45	49
	Non	26	2111

R=36.0%
P=76.2%
F_1=48.9%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	32	10
	Non	57	2208

R=39.3%
P=48.9%
F_1=43.6%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	22	23
	Non	34	2225

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2296	491
	Non	491	20017

Tableaux 6.13 : Matrices de contingence 9 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 9 agents est

F_1=82.4%

- Dérouler la même démarche par 21 agents

R=95.1%
P=94.0%
F_1=94.6%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1034	66
	Non	53	1214

R=93.2%
P=92.8%
F_1=93.0%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	670	52
	Non	49	1579

R=78.8%
P=77.5%
F_1=78.1%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	141	41
	Non	38	2104

R=48.3%
P=57.1%
F_1=52.4%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	72	54
	Non	77	2154

R=56.6%
P=82.9%
F_1=67.3%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	22
	Non	82	2132

R=95.7%
P=53.8%
F_1=68.9%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	112	96
	Non	5	2137

R=71.8%
P=67.6%
F_1=69.6%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	94	45
	Non	37	2128

R=66.2%
P=51.1%
F_1=57.7%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	47	45
	Non	24	2100

R=39.3%
P=79.5%
F_1=52.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	35	9
	Non	54	2201

R=44.6%
P=55.6%
F_1=49.5%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	25	20
	Non	31	2117

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2337	450
	Non	450	19866

Tableaux 6.14 : Matrices de contingence 21 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 21 agents est

F_1=83.9%

- Dérouler une autre fois la même démarche par un SMA de 33 agents

R=95.8%
P=94.6%
F_1=95.2%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1041	59
	Non	46	1220

R=92.8%
P=92.4%
F_1=92.6%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	667	55
	Non	52	1584

R=80.4%
P=77.4%
F_1=78.9%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	144	42
	Non	35	2112

R=53.0%
P=64.8%
F_1=58.3%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	79	43
	Non	70	2142

R=59.8%
P=87.6%
F_1=71.1%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	113	16
	Non	76	2128

R=97.4%
P=54.8%
F_1=70.2%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	114	94
	Non	3	2141

R=74.8%
P=70.5%
F_1=72.6%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	98	41
	Non	33	2136

R=73.2%
P=56.5%
F_1=63.8%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	52	40
	Non	19	2096

R=42.7%
P=86.4%
F_1=57.1%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	38	6
	Non	51	2196

R=50.0%
P=62.2%
F_1=55.4%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	28	17
	Non	28	2108

Matrice de contingence globale du corpus		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	2374	413
	Non	413	19863

Tableaux 6.15 : Matrices de contingence 33 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 33 agents est

$$F_1=85.2\%$$

- Recommencer les mêmes traitements avec un SMA de 61 agents

R=96.0%
P=64.9%
$F_1=95.5\%$

Catégorie Earn C1		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	1044	56
	Non	43	1218

R=92.9%
P=92.5%
$F_1=92.7\%$

Catégorie Acquisition C2		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	668	54
	Non	41	1590

R=81.6%
P=78.5%
$F_1=80.0\%$

Catégorie Money-fx C3		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	146	40
	Non	33	2111

R=55.7%
P=68.0%
$F_1=61.3\%$

Catégorie Grain C4		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	83	39
	Non	66	2139

R=60.3%
P=88.4%
$F_1=71.7\%$

Catégorie Crude C5		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	114	15
	Non	75	2122

R=99.1%
P=55.8%
$F_1=71.4\%$

Catégorie Trade C6		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	116	92
	Non	1	214

R=77.1%
P=72.7%
F_1=74.8%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	101	38
	Non	30	2125

R=77.5%
P=59.8%
F_1=67.5%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	55	37
	Non	16	2084

R=46.1%
P=93.2%
F_1=61.7%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	41	3
	Non	48	2184

R=55.4%
P=68.9%
F_1=61.4%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	31	14
	Non	25	2102

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2399	388
	Non	388	17889

Tableaux 6.16 : Matrices de contingence 61 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 61 agents est

F_1=86.1%

- Encore une fois avec 99 agents

R=94.3%
P=93.0%
F_1=93.7%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1025	77
	Non	62	1218

R=91.7%
P=91.4%
F_1=91.5%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	659	62
	Non	60	1595

R=78.2%
P=78.2%
F_1=78.2%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	140	39
	Non	39	2114

R=52.3%
P=60.9%
F_1=56.3%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	78	50
	Non	71	2142

R=56.6%
P=82.9%
F_1=67.3%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	22
	Non	82	2129

R=93.2%
P=52.4%
F_1=67.1%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	109	99
	Non	8	227

R=74.8%
P=70.5%
F_1=72.6%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	98	41
	Non	33	2127

R=73.2%
P=56.5%
F_1=63.8%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	52	40
	Non	19	2088

R=43.8%
P=88.6%
F_1=58.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	39	5
	Non	50	2200

R=51.8%
P=64.4%
F_1=57.4%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	29	16
	Non	27	2118

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2336	451
	Non	451	17958

Tableaux 6.17 : Matrices de contingence 99 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 99 agents est

$$F_1=83.8\%$$

- Et enfin répéter le processus par un SMA de 181 agents

R=92.5%
P=91.3%
$F_1=91.9\%$

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1006	96
	Non	81	1233

R=88.6%
P=88.6%
$F_1=88.6\%$

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	637	82
	Non	82	1603

R=70.9%
P=70.6%
$F_1=70.8\%$

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	127	53
	Non	52	2138

R=49.7%
P=57.4%
$F_1=53.2\%$

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	74	55
	Non	75	2145

R=51.3%
P=75.2%
$F_1=61.0\%$

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	97	32
	Non	92	2141

R=86.3%
P=48.6%
$F_1=62.2\%$

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	101	107
	Non	16	248

R=69.5%
P=65.5%
$F_1=67.4\%$

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	91	48
	Non	40	2138

R=66.2%
P=51.1%
F_1=57.7%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	47	45
	Non	24	2095

R=37.1%
P=75.0%
F_1=49.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	33	11
	Non	56	2214

R=41.1%
P=51.1%
F_1=45.5%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	23	22
	Non	33	2127

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2236	551
	Non	551	18082

Tableaux 6.18 : Matrices de contingence 181 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 181 agents est

F_1=80.2%

- **Fixer le nombre d'agents composant notre SMA**

  Le modèle de classification de textes construit à base d'un SMA constitué de 61 agents, offrant les résultats optimaux, sera adopté pour la classification finale.

6.5.4- Comparaison des résultats

Nous proposons une série de comparaisons en quatre étapes :

1- En mettant en compétitions en premier temps nos résultats obtenus par les 6 variantes de Naïve Bayes (Documents représentés en 2, 3, 4, 5, 6 et 7-grammes).

2- Dans la deuxième comparaison nous allons confronter le meilleur résultat fourni par nos 6 modèles précédents avec des résultats références de 6 méthodes de catégorisation dans le domaine, à savoir les machines à vecteurs supports, Rocchio, les plus proches voisins, les arbres de décision, Naïve Bayes et les réseaux de neurones, appuyés sur les résultats obtenus par (Dumais & all, 1998), (Joachims, 1998), (Yang & Liu, 1999) et (Li & Yang, 2003), confrontés dans plusieurs études comme dans (Sebastiani, 2002), (Nakache, 2007) et (Manning & all, 2008), pour se situer dans une échelle des spécialistes du domaine et donner à nos performances une certaine crédibilité.

3- La troisième consiste à comparer les résultats obtenus en Mono-Agent avec ceux des SMA en renforçant chaque fois le nombre d'agents (3, 9, 21, 33, 61, 99 jusqu'à 181 agents).

4- La dernière comparaison va mettre en oppositions tous les six classifieurs ajouté à notre classifieur Naïve Bayes (Approche non distribuée) avec notre nouveau modèle Naïve Bayes basée sur une approche distribuée (SMA) composé de 61 agents.

6.5.4.1- Comparaison des résultats obtenus aves différentes valeurs de N (N-grammes)

Ce tableau reflète les résultats obtenus par l'algorithme Naïve Bayes avec des textes codés en 2, 3, 4, 5, 6 et 7-grammes :

F_1	N=2	N=3	N=4	N=5	N=6	N=7
Earn	93.5%	**93.6%**	92.7%	85.4%	82.9%	79.9%
Acq	67.2%	87.3%	**90.3%**	86.8%	84.4%	82.6%
money-fx	31.2%	52.2%	**75.1%**	73.5%	71.4%	67.1%
Grain	7.5%	35.1%	45.2%	55.8%	**59.7%**	55.1%
Crude	6.1%	56.3%	65.3%	**67.7%**	63.2%	52.1%
trade	28.2%	**68.2%**	60.9%	53.1%	50.3%	46.6%
Interest	31.3%	65.0%	**68.9%**	58.5%	52.5%	51.0%
ship	42.5%	**48.7%**	48.5%	39.4%	2.7%	2.6%
wheat	25.9%	**46.8%**	41.2%	38.8%	39.1%	38.0%
corn	28.2%	34.9%	**36.0%**	3.5%	11.8%	14.8%
Micro-Avg	**60.6%**	**77.5%**	**80.6%**	**75.5%**	**73.0%**	**70.3%**

Tableau 6.19 : Comparaison des résultats obtenus avec les différents N (N-grammes)

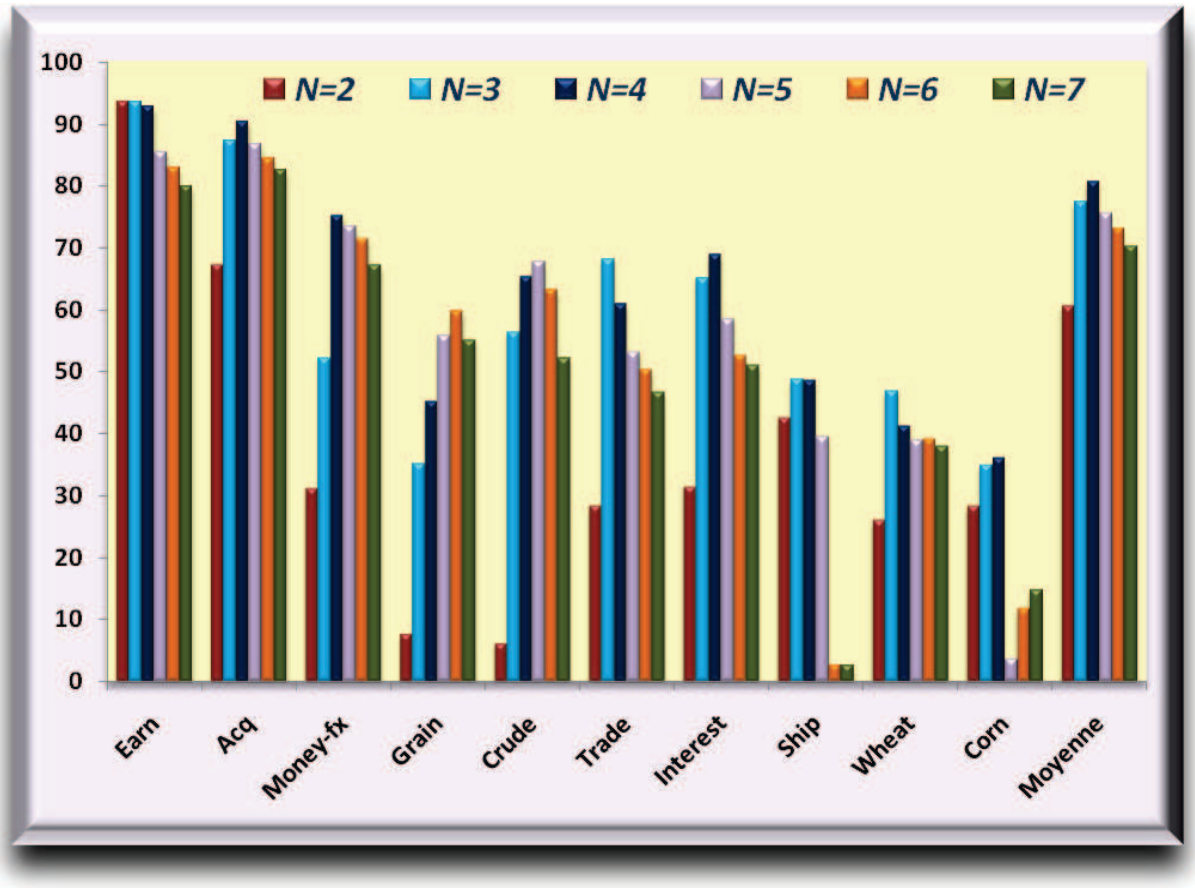

Figure 6.2: Comparaison des résultats obtenus avec les différents N (N-grammes)

6.5.4.2- Comparaison des résultats d'autres algorithmes

Le résultat de comparaison des méthodes SVM, Rocchio, kNN, les arbres de décision, Naïve Bayes et les réseaux de neurones avec notre algorithme Naïve Bayes (4-grammes) est le suivant :

F_1	SVM	Rocchio	kNN	Arb.Déc	Rés.Neur	N.Bayes	Notre NB
Earn	98.0%	92.9%	96.7%	97.8%	94.1%	95.9%	**92.7%**
Acq	93.6%	64.7%	91.6%	89.7%	88.8%	87.8%	**90.3%**
Money-fx	74.5%	46.7%	78.0%	66.2%	74.2%	56.6%	**75.1%**
Grain	94.6%	67.5%	86.4%	85.0%	73.8%	78.8%	**45.2%**
Crude	88.9%	70.1%	87.4%	85.0%	86.5%	79.5%	**65.3%**
Trade	75.9%	65.1%	77.3%	72.5%	79.5%	63.9%	**60.9%**
Interest	77.7%	63.4%	73.7%	67.1%	83.9%	64.9%	**68.9%**
Ship	85.6%	49.2%	49.4%	74.2%	89.9%	85.4%	**48.5%**
Wheat	91.8%	68.9%	69.1%	92.5%	79.7%	69.7%	**41.2%**
Corn	90.3%	48.2%	48.5%	91.8%	77.2%	65.3%	**36.0%**
Micro-Avg	**92.0%**	**64.6%**	**81.8%**	**88.4%**	**82.8%**	**81.5%**	**80.6%**

Tableau 6.20 : Comparaison des résultats obtenus avec ceux des autres algorithmes

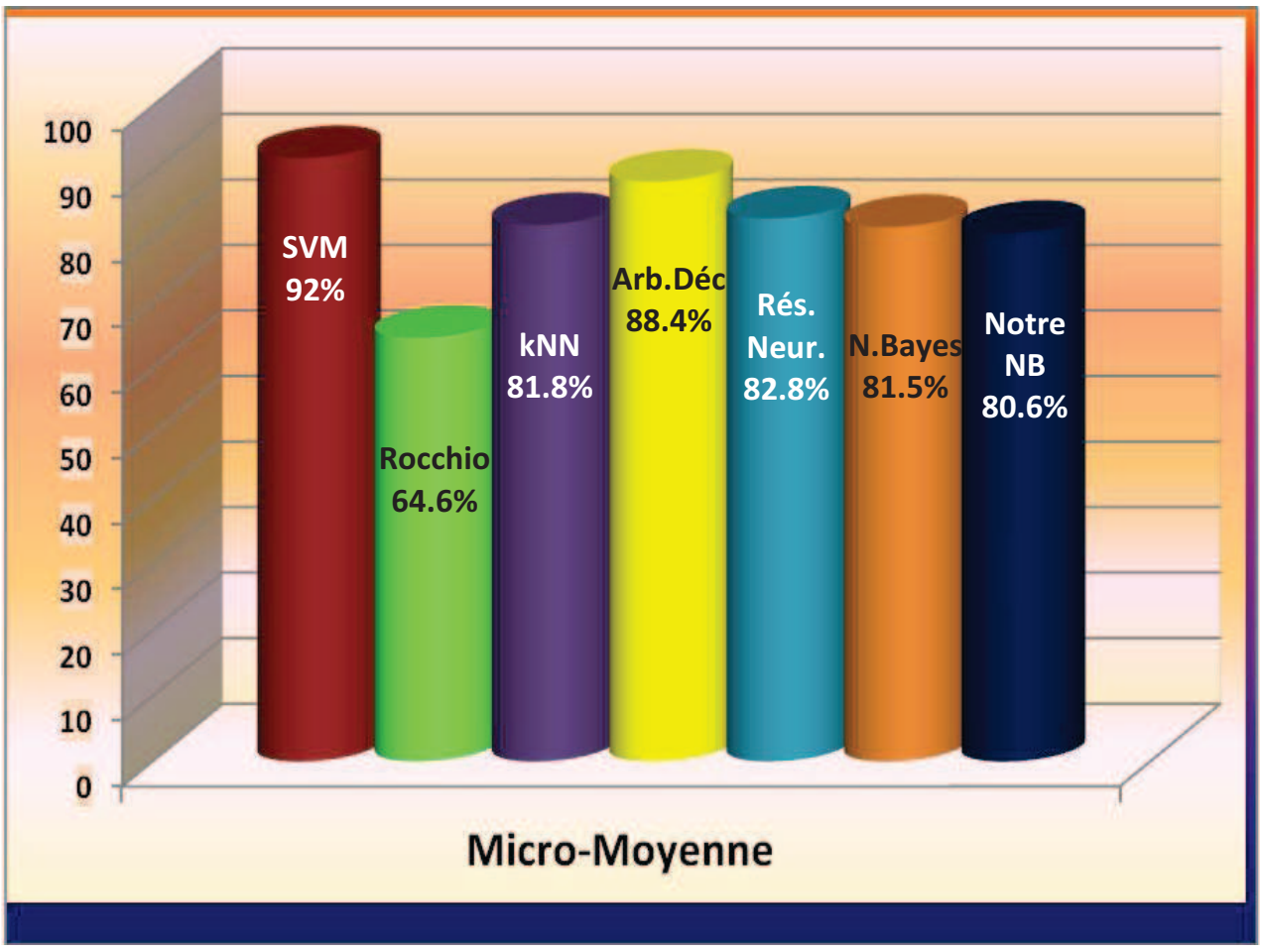

Figure 6.3 : Comparaison des résultats obtenus avec ceux des différents algorithmes

6.5.4.3- Comparaison des approches Mono et Multi-Agents en variant le nombre d'agents

Les deux tableaux suivants opposent tous les résultats obtenus avec le même algorithme Naïve Bayes avec des textes codés en 4-grammes, en commençant par le Mono-Agent et en augmentant au fur et à mesure le nombre d'agents. Les comparaisons vont être appuyées sur deux critères principaux à savoir les performances du classifieur en qualité des ses résultats et son efficacité en temps d'exécution du processus (prétraitement, apprentissage et test) sur tout le corpus.

F_1	MonoAgent	3Agents	9Agents	21Agents	33Agents	61Agents	99Agents	181Agents
Earn	92,7%	93,2%	93,6%	94,6%	95,2%	**95,5%**	93,7%	91,9%
Acq	90,3%	90,9%	92,2%	93,0%	92,6%	**92,7%**	91,5%	88,6%
Money-fx	75,1%	73,5%	74,2%	78,1%	78,9%	**80,0%**	78,2%	70,8%
Grain	45,2%	45,7%	46,5%	52,4%	58,3%	**61,3%**	56,3%	53,2%
Crude	65,3%	66,4%	68,6%	67,3%	71,1%	**71,7%**	67,3%	61,0%
Trade	60,9%	63,9%	66,7%	68,9%	70,2%	**71,4%**	67,1%	62,2%
Interest	68,9%	67,9%	70,2%	69,6%	72,6%	**74,8%**	72,6%	67,4%
Ship	48,5%	49,1%	54,5%	57,7%	63,8%	**67,5%**	63,8%	57,7%
Wheat	41,2%	46,6%	48,9%	52,6%	57,1%	**61,7%**	58,6%	49,6%
Corn	36,0%	41,2%	43,6%	49,5%	55,4%	**61,4%**	57,4%	45,5%
Micro-Avg	**80,6%**	**81,2%**	**82,4%**	**83,9%**	**85,2%**	**86,1%**	**83,8%**	**80,2%**

Tableau 6.21 : Comparaison des résultats obtenus avec les différents nombres d'agents

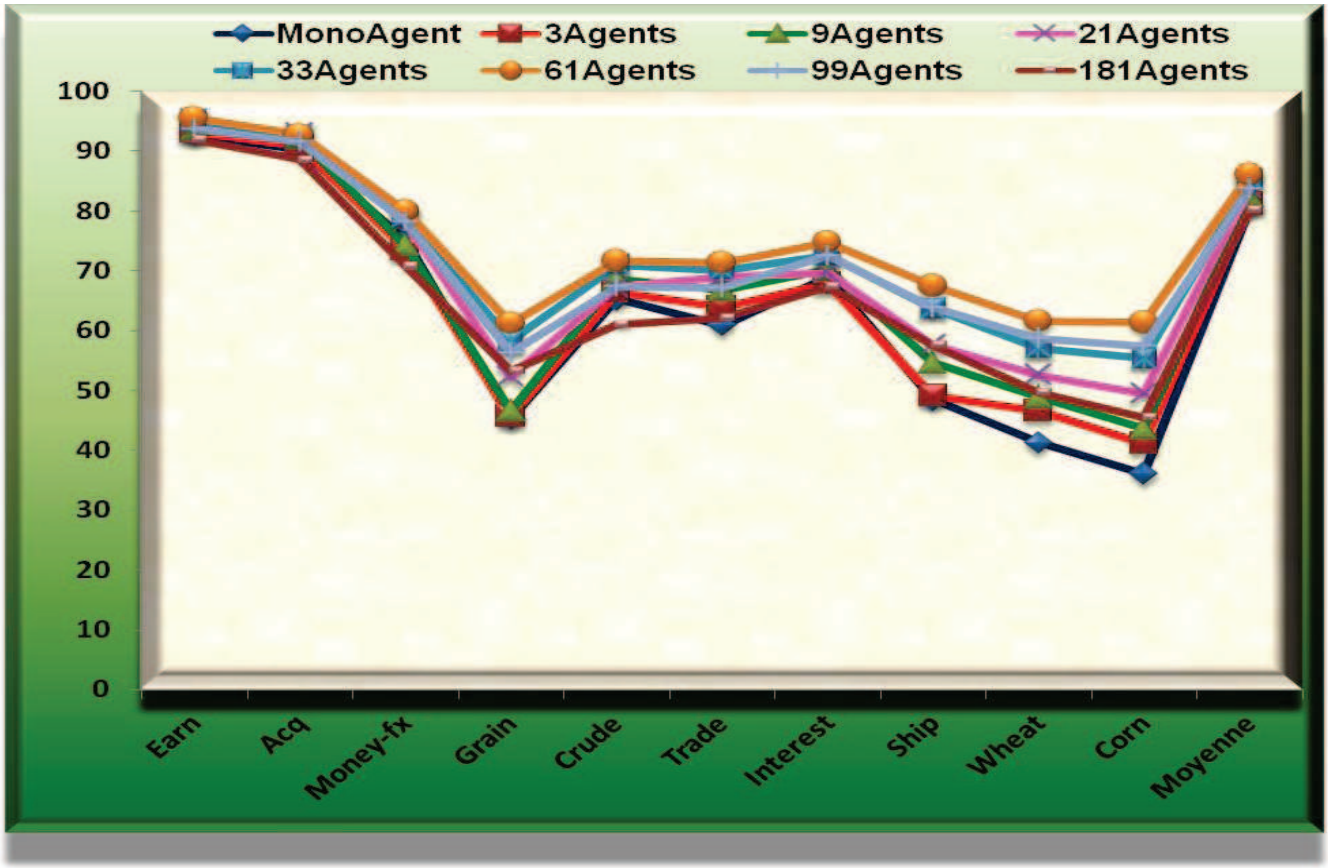

Figure 6.4 : Comparaison des résultats obtenus avec les différents nombres d'agents

Corpus	MonoAgent	3Agents	9Agents	21Agents	33Agents	61Agents	99Agents	181Agents
Prétrait.	334 Min	169 Min	75 Min	36 Min	30 Min	24 Min	27 Min	29 Min
App+Test	3 Min	8 Min	19 Min	37Min	53 Min	70 Min	105 Min	149 Min
Temps Exéc	**337 Min**	**177 Min**	**94 Min**	**73 Min**	**83 Min**	**94 Min**	**132 Min**	**178 Min**

Tableau 6.22 : Evaluation des temps d'exécution des systèmes mono et multi-agents

N.B : Les temps d'exécution sont évalués sur un HP Compaq DX7500, Pentium(R) Dual-Core CPU 2.5Ghz, 3 Go de mémoire vive.

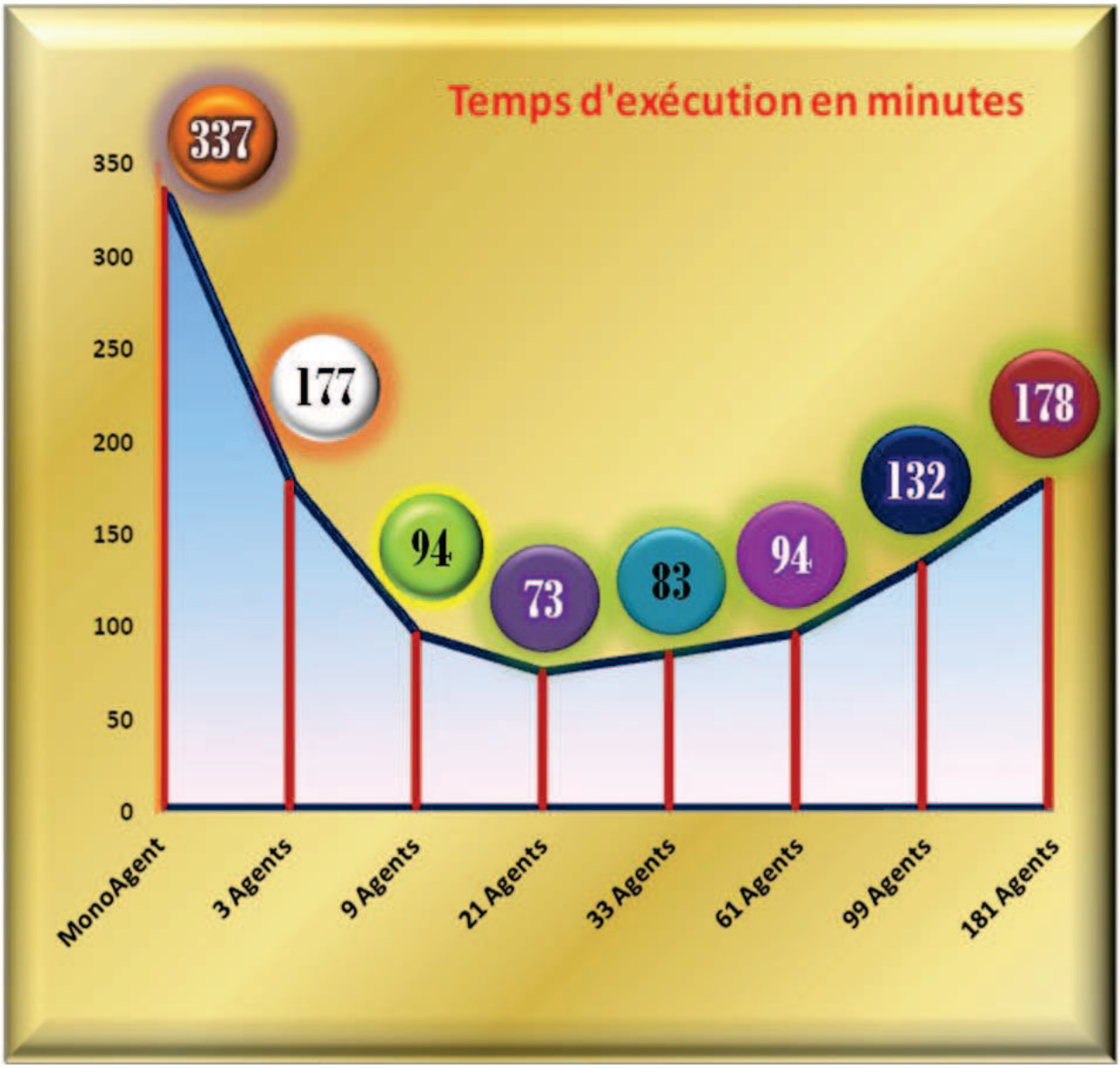

Figure 6.5 : Evaluation des temps d'exécution des systèmes mono et multi-agents

6.5.4.4- Comparaison des approches non distribuées avec notre approche SMA

La dernière confrontation va mettre en compétition les six classifieurs ajouté à notre classifieur Naïve Bayes Mono-Agent avec notre nouveau modèle construit Naïve Bayes basée sur une approche distribuée (SMA) composé de 61 agents.

F_1	SVM	Rocchio	kNN	Arb.Déc	Rés.Neur	N.Bayes	Notre NB	NB(SMA)
Earn	98.0%	92.9%	96.7%	97.8%	94.1%	95.9%	92.7%	95,5%
Acq	93.6%	64.7%	91.6%	89.7%	88.8%	87.8%	90.3%	92,7%
Money-fx	74.5%	46.7%	78.0%	66.2%	74.2%	56.6%	75.1%	80,0%
Grain	94.6%	67.5%	86.4%	85.0%	73.8%	78.8%	45.2%	61,3%
Crude	88.9%	70.1%	87.4%	85.0%	86.5%	79.5%	65.3%	71,7%
Trade	75.9%	65.1%	77.3%	72.5%	79.5%	63.9%	60.9%	71,4%
Interest	77.7%	63.4%	73.7%	67.1%	83.9%	64.9%	68.9%	74,8%
Ship	85.6%	49.2%	49.4%	74.2%	89.9%	85.4%	48.5%	67,5%
Wheat	91.8%	68.9%	69.1%	92.5%	79.7%	69.7%	41.2%	61,7%
Corn	90.3%	48.2%	48.5%	91.8%	77.2%	65.3%	36.0%	61,4%
Micro-Avg	**92.0%**	**64.6%**	**81.8%**	**88.4%**	**82.8%**	**81.5%**	**80.6%**	**86,1%**

Tableau 6.23 : Comparaison des différents résultats avec l'approche distribuée

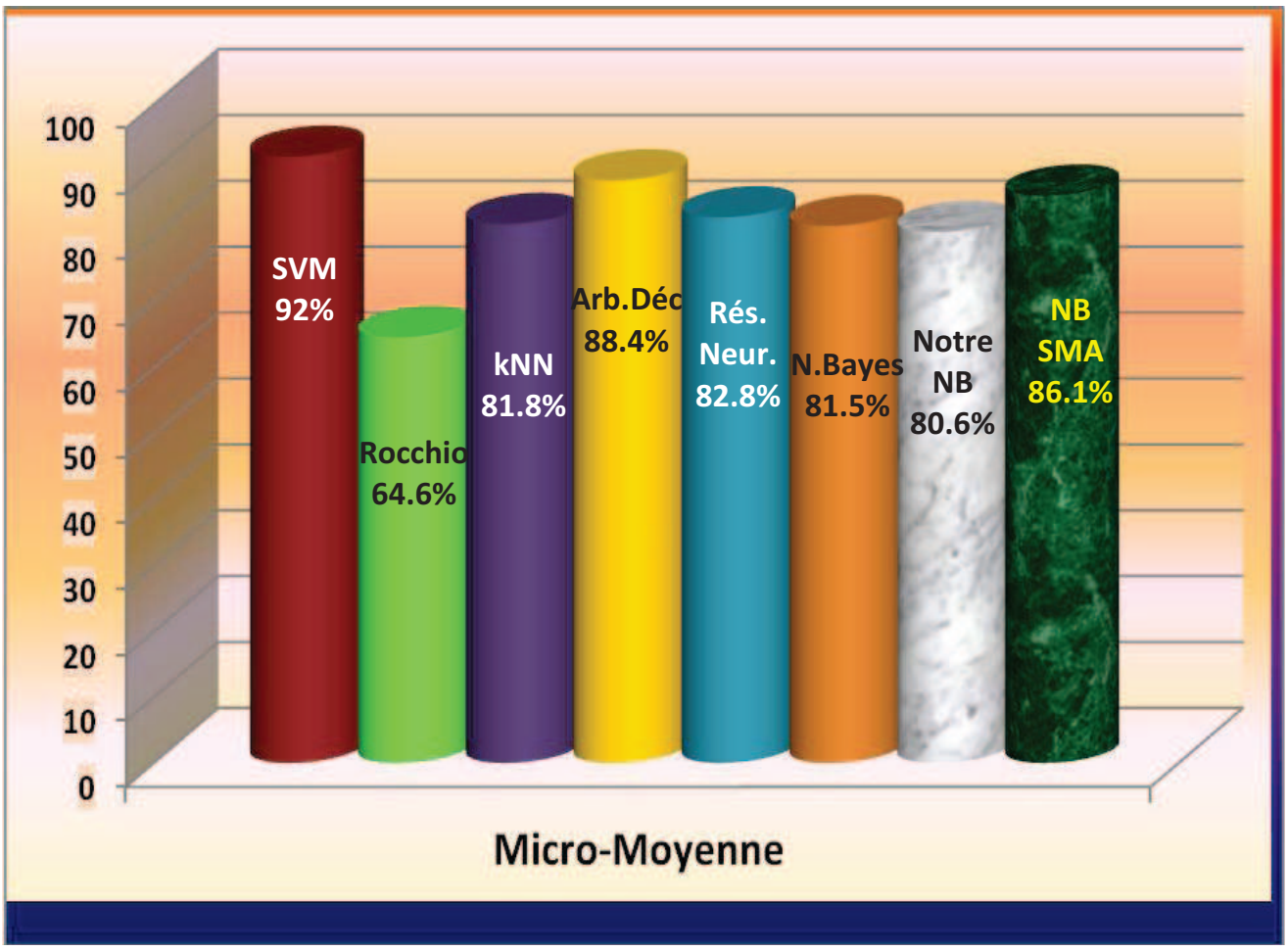

Figure 6.6 : Comparaison des différents résultats avec l'approche distribuée

6.6- Discussion

6.6.1- L'influence du N dans les résultats de l'approche

La représentation basée sur les N-grammes est dépendante d'un paramètre essentiel : la valeur de N, c.-à-d. le nombre de caractères que contiendra chaque N-grammes.
Qu'elle est la valeur de N qui donne les meilleurs résultats ?
Pour répondre à cette question, nous avons appliqué l'approche NB sur le corpus Reuters pour des valeurs de N comprise entre 2 et 7.
Le tableau 6.19 et la figure 6.2 Présentent les résultats obtenus en utilisant la mesure de performance F_1.
En analysant les résultats du tableau, on remarque que les performances s'améliorent en accroissant la valeur de N jusqu'à N=4 qui présente la valeur optimale. Ces performances commencent à rechuter à partir de N=7.

6.6.2- L'influence du nombre d'agents dans les résultats de classification

La variation du nombre d'agents était une expérience très intéressante. Après les différentes expérimentations sur 3, 9, 21, 33, 61, 99 et 181 agents, les résultats commencent à se stabiliser à partir de 61 agents. Ainsi une forte distribution sur un grand nombre d'agents n'est pas nécessaire, puisque que les résultats à partir de 61 agents commencent à rechuter légèrement. Ce qui nous amène à conclure qu'une Soixantaine d'agents est très satisfaisante pour un système de catégorisation automatique de textes.
Les différentes mesures F_1 obtenus en variant le nombre d'agents sont exposées dans le tableau 6.21 et la figure 6.4.

6.6.3- L'apport de la distribution de classification

Le développement d'un modèle fondé sur une architecture multi-agents a porté ses fruits puisque les résultats obtenus par notre modèle SMA sont nettement meilleurs du modèle mono-agent, mais sans autant abuser dans la distribution car une forte distribution va générer des vocabulaires assez pauvres en pouvoir informatif qui va certainement engendrer une dégradation considérable dans la qualité des résultats, d'une part (tableau 6.23 et figure 6.6).
D'autre part, l'amélioration en matière d'efficacité du classifieur à savoir le temps d'exécution du processus pour accomplir les différentes fonctions de prétraitement, apprentissage et test, est remarquable puisque les résultats figurés dans le tableau 6.22 et la figure 6.5 sont considérables.
Toutefois nous tenons à signaler que la distribution a influencé sur le temps d'exécution du prétraitement et de l'apprentissage très favorablement contrairement au test ou la distribution a augmenté légèrement le temps d'exécution puisque tous les agents vont tester les mêmes documents du corpus (2788 documents) chacun à son tour sachant en fin que l'opération de prétraitement est plus exigeante en temps d'exécution que le test.
Pour que nos comparaisons, des différents classifieurs Mono et Multi-Agents, en temps d'exécution soient effectives, nous avons comptabilisé le temps global d'exécution du processus (prétraitement + apprentissage + test).
Si on prend en considération le critère vitesse d'exécution, évidemment le modèle à 21 agents est le plus rapide mais avec un taux de classification pas aussi performant que le modèle à 33 ou 61 agents, sachant que ce dernier critère, à savoir la qualité de résultats de notre classifieur, est supposé la première priorité de nos études. Sans oublier à rappeler également, que les opérations de prétraitements, apprentissage et test se font en offline, donc le temps d'exécution ne sera d'une importance que si on veut refaire ces opérations ou traiter un autre

corpus. D'ailleurs, l'objectif majeur ciblé dans cette étude était la recherche du meilleur compromis qualité/efficacité.
Finalement après une lecture et une synthèse faite des résultats, le modèle SMA à 61 agents s'illustre comme le modèle optimal pour une classification automatique distribuée de Naïve Bayes de documents représentés en 4-Grammes.

6.7- Conclusion

Au cours de ce chapitre, nous avons présenté l'approche proposée avec toutes ces étapes, une approche qui tire son profit de l'utilisation des n-grammes comme méthode pour représenter les textes, et de l'algorithme Naïve Bayes comme algorithme d'apprentissage mais surtout l'apport considérable de notre approche c'est la distribution du processus de classification basée sur le paradigme agent.

Nous avons analysé les résultats de cette approche sur le corpus Reuters21578-Top10.

Les expérimentations réalisées ont mené aux constatations suivantes :

1- Le choix de la valeur N, influence sur les résultats de l'approche. En effet, les expérimentations ont montré que les quint-grams sont idéales pour le corpus. Cette valeur peut changer pour d'autres corpus.

2- En général, Les modèles Naïve Bayes classent bien dans le domaine textuel, et en particulier notre classifieur à base des N-Grammes a amené à des résultats très encourageants (80.6 %), mais insuffisants à l'égard des méthodes connues dans la littérature par la qualité de leurs résultats.

3- La nouvelle utilisation du classifieur Naïve Bayes, basée sur une architecture multi-agents introduite dans notre approche, a atteint l'objectif tracé puisque nous avons amélioré considérablement les performances du modèle basé sur un seul module logiciel (+ 5.5%), en s'approchant nettement des meilleurs résultats obtenus dans la littérature sur Reuters Top10 à savoir les SVM et Arbres de Décision.

4- Un autre atout dans la distribution de classification est en matière d'efficacité du classifieur qui s'améliore très nettement, bien sûr sans exagérer dans la distribution.