Classification automatique de documents bruités à faible contenu textuel

Sami Laroum, Nicolas Béchet, Hatem Hamza, Mathieu Roche

Classification automatique de documents bruités à faible contenu textuel

Sami Laroum*, Nicolas Béchet**, Hatem Hamza*** et Mathieu Roche**

* Biopolymères Intéractions Assemblages, INRA
BP 71627, 44316 Nantes - France
sami.laroum@nantes.inra.fr,
** LIRMM Université Montpellier II CNRS
161 Rue Ada, 34392 Montpellier -France
{nicolas.bechet, Mathieu.Roche}@lirmm.fr,
http://www.lirmm.fr/
***ITESOFT: Parc d'Andron
Le Séquoia 30470 Aimargues -France
Hatem.Hamza@itesoft.com
http://www.itesoft.fr/

Résumé. La classification de documents numériques est une tâche complexe dans un flux numérique de gestion électronique de documents. Cependant, la quantité des documents issus de la retro-conversion d'OCR (Reconnaissance Optique de Caractères) constitue une problématique qui ne facilite pas la tâche de classification. Après l'étude et l'évaluation des descripteurs les mieux adaptés aux documents issus d'OCR, nous proposons une nouvelle approche de représentation des données textuelles : l'approche HYBRED (**HYB**rid **RE**presentation of **D**ocuments). Elle permet de combiner l'utilisation de différents descripteurs d'un texte afin d'obtenir une représentation plus pertinente de celui-ci. Les expérimentations menées sur des données réelles ont montré l'intérêt de notre approche.

1 Introduction

Aujourd'hui, nous vivons dans un monde où l'information est disponible en grande quantité tout en étant de qualité très diverse. Internet s'enrichit continuellement de nouveaux contenus. Par exemple, les entreprises emmagasinent de plus en plus de données, le courriel devient un moyen de communication extrêmement populaire, des documents autrefois manuscrits sont aujourd'hui disponibles sous format numérique. Mais toute cette information complexe serait sans intérêt si notre capacité à y accéder efficacement n'augmentait pas elle aussi. Pour cela, nous avons besoin d'outils permettant de chercher, classer, conserver, mettre à jour et analyser les données accessibles. Il est ainsi nécessaire de proposer des systèmes afin d'accéder rapidement à l'information désirée, réduisant ainsi l'implication humaine.

Un des domaines qui tente d'apporter des améliorations et de réduire la tâche de l'humain est la classification automatique de documents. Celle-ci consiste à associer une catégorie à un

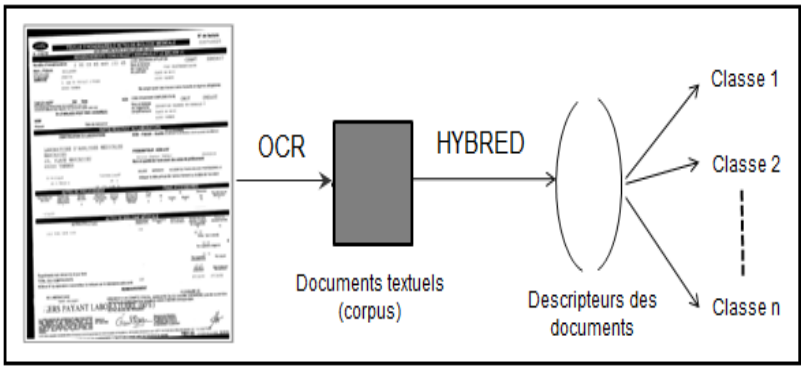

FIG. 1 – *Processus de classification.*

document pouvant être une phrase, un paragraphe, un texte, etc.
Généralement, une classification de documents complexes est effectuée manuellement et sa réalisation est donc coûteuse en terme de temps. En effet, chaque texte (ou une partie) doit être manuellement lu pour attribuer une catégorie adaptée (classe). C'est la raison pour laquelle le domaine de la classification automatique de documents est en perpétuel développement.

Dans cette étude conduite pour améliorer les performances de la classification automatique de documents textuels issus d'OCR (Reconnaissance Optique de Caractères), nous proposons d'évaluer la pertinence de différents descripteurs de données textuelles complexes, au regard du bruit présent et du faible contenu textuel. Junker et Hoch (1997) traitent la même problématique en évaluant des documents OCR par l'application de différentes représentations afin de déterminer les plus pertinentes. Nos travaux reposent sur l'évaluation d'un nombre plus important de descripteurs, avec notamment l'utilisation d'un filtrage grammatical pour finalement présenter une approche hybride nommée **HYBRED** (**HYB**rid **RE**presentation of **D**ocuments). Celle-ci combine les "meilleurs" descripteurs de données complexes issues d'OCR. La figure 1 présente l'architecture générale d'une tâche de classification automatique de données dans notre contexte.

Afin de tester la pertinence de notre approche, nous nous appuyons sur des corpus de la société ITESOFT[1]. Les documents de ces corpus sont répartis dans différentes classes telles que *attestation salaire*, *facture optique* ou encore *frais médecin*. Ces corpus ont la particularité de provenir de rétro-conversion d'OCR, engendrant une quantité non négligeable de bruit, pouvant notamment être des fautes d'orthographe, des lettres manquantes, etc. Une telle situation complique la tâche de classification.

L'article est organisé de la manière suivante : la section 2 décrit un processus global de classification automatique de documents. Nous présentons ensuite un état de l'art des différentes méthodes de représentation de données et principalement celles que nous allons utiliser. Notre approche HYBRED est décrite dans la section 4. La section 5 présente les différentes expérimentations réalisées ainsi que les résultats obtenus avec les différents descripteurs pour finalement conclure en section 6.

[1] http ://www.itesoft.fr/

2 Processus de classification

Dans notre approche, deux étapes sont clairement identifiées. L'étape de représentation des données et la classification de documents.

2.1 Représentation vectorielle des documents textuels

L'exploitation et le traitement automatique des documents, en particulier pour les tâches de classification, nécessitent une première étape consistant à les représenter. Pour cela, la méthode la plus courante consiste à projeter les données textuelles (par exemple les mots) dans un espace vectoriel. De nombreux travaux utilisent une telle approche. Citons par exemple Vinot et al. (2003) qui représentent les données à l'aide d'un modèle vectoriel pour une tâche de classification de documents à contenus racistes. Memmi (2000) et Pouliquen et al. (2002) utilisent quant à eux un modèle vectoriel pour calculer des scores de similarité entre différents documents. Les travaux de Pisetta et al. (2006) emploient des techniques d'analyse linguistique pour extraire un ensemble de termes candidats qui permettront de construire des concepts relatifs au corpus étudié. Ensuite, ils utilisent un modèle vectoriel pour représenter les documents par un vecteur de concepts.

Le modèle vectoriel le plus couramment utilisé dans la littérature est le "Vector Space Model" de Salton et al. (1975) qui permet d'obtenir les mêmes performances qu'une indexation manuelle. Le modèle de Salton consiste à représenter un corpus par une matrice telle que les lignes soient relatives aux descripteurs et les colonnes aux documents. Une cellule d'une telle matrice comptabilise la fréquence d'apparition d'un descripteur dans un document. Ainsi, la matrice formée peut être utilisée pour effectuer diverses tâches automatiques de fouille de textes.

2.2 Phase de classification

Le principe d'une classification automatique de textes est d'utiliser un modèle afin de classer un document dans une catégorie pertinente. Nous pouvons distinguer deux types de modèles de classification : ceux nécessitant une phase d'apprentissage et ceux sans phase d'apprentissage. Parmi les modèles utilisant un apprentissage, nous distinguons l'apprentissage supervisé et non supervisé. Nous effectuons dans nos travaux uniquement de la classification avec apprentissage supervisé.

La notion d'**apprentissage** introduit le fait d'*apprendre* un ensemble de relations entre les critères caractérisant l'élément à classer et sa classe cible. Les algorithmes de classification avec apprentissage ont recours à un ensemble d'exemples afin d'apprendre ces relations. La notion de **supervisé** signifie que les exemples sont étiquetés (la classe est connue).

Il existe de nombreuses méthodes de classification avec apprentissage supervisé. Dans cet article, nous allons nous appuyer sur les algorithmes suivants qui sont les plus utilisés dans la littérature :

– ***k* Plus Proches Voisins (k-PPV)** : k-PPV est un algorithme qui a montré son efficacité face au traitement de données textuelles (Yang (1999)). Le classement d'un nouveau texte (entrée x) s'opère en effectuant un calcul de similarité (distance euclidienne, cosinus, etc.) entre la représentation vectorielle d'un nouveau document et celles des

exemples du corpus. La méthode consiste alors à calculer et classer les scores de similarité obtenus par ordre décroissant et ne garder que les k premiers. La classe majoritaire parmi ces k documents est alors attribuée au nouveau document. Donc, comme son nom l'indique, la classification d'un nouveau document est fonction de ses k voisins les plus proches.

– **Naïve Bayes** : Le classifeur de type Naïve Bayes est un catégoriseur de type probabiliste fondé sur le théorème de Bayes (1763). Considérons $v_j = (v_{j1}, ..., v_{jk}, ..., v_{jd})$ un vecteur de variables aléatoires représentant un document d_j et C un ensemble de classes. En s'appuyant sur le théorème de Bayes, la probabilité que ce dernier appartienne à la classes $c_i \in C$ est définie par :

$$P(c_i|v_j) = \frac{P(c_i)P(v_j|c_j)}{P(v_j)}$$

La variable aléatoire v_{jk} du vecteur v_j représente l'occurrence de l'unité linguistique k retenue pour la classification dans le document d_j.
La classe c_k d'appartenance de la représentation vectorielle v_j d'un document d_j est définie comme suit :

$$c_k = arg\ max\ P(c_i \in C) \prod_k P(v_{jk}|c_j)$$

En d'autres termes, le classificateur Naïve Bayes affecte au document d_j la classe ayant obtenue la probabilité d'appartenance la plus élevée.
Alors, $p(c_i)$ est définie de la façon suivante :

$$P(c_i) = \frac{nombre\ de\ documents \in c_i}{nombre\ total\ de\ documents}$$

En faisant l'hypothèse que les v_j sont indépendantes, la probabilité conditionnelle $P(v_j|c_i)$ est définie ainsi :

$$P(v_j|c_i) = P(v_{jk}|c_i)$$

Une telle hypothèse d'indépendance des v_j peut néanmoins dégrader qualitativement les résultats obtenus avec une telle approche (Lewis (1998)).

– **Machines à support de vecteurs (SVM)** : L'algorithme des SVM est originalement un algorithme mono-classe permettant de déterminer si un élément appartient (qualifié de positif) ou non (qualifié de négatif) à une classe. L'idée principale de cet algorithme est de trouver un hyperplan qui sépare au mieux les exemples positifs des exemples négatifs en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Intuitivement, cela garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan, mais être situés franchement d'un côté ou de l'autre de la frontière. SVM est considéré comme un des algorithmes les plus performants en classification textuelle (Joachims (1998)). Dans nos travaux, nous utilisons une extension de l'algorithme original des SVM permettant la gestion multi-classes, l'algorithme SMO (Platt (1999)).

Une évaluation rigoureuse des différentes méthodes de classification avec apprentissage supervisé s'appuie généralement sur l'utilisation du processus de "**validation croisée**". Elle consite à segmenter le corpus initial en x parties de même taille. En général, le nombre x de parties est fixé à dix : neuf parties pour l'apprentissage et une de test. Ainsi, les différents documents constituant le corpus deviennent alternativement corpus de test et d'apprentissage. Une telle méthode permet d'avoir une robustesse dans le choix de l'algorithme ou des paramètres.

3 Travaux antérieurs relatifs à la représentation des documents

Comme nous l'avons précisé précédemment, les corpus que nous manipulons sont issus de la rétroconversion d'OCR. La principale caractéristique du processus d'OCR que nous appliquons tient dans la diversité des données traitées ainsi qu'à leur faible contenu textuel. Par ailleurs, ces corpus sont syntaxiquement pauvres (peu de phrases bien formulées en langage naturel) et bruités. Dans notre contexte, le choix des descripteurs peut se révéler décisif. Nous allons utiliser différents descripteurs pour la tâche de classification à partir de données pauvres et bruitées. Nous rappelons que notre approche qui sera présentée en section 4 prendra en compte les descripteurs les plus adaptés dans notre contexte de classification de données bruitées.

3.1 Les mots

La représentation des documents d'un corpus par des mots consiste à utiliser les mots sous leur forme originale comme descripteur pour ensuite représenter les textes sous forme vectorielle. Un prétraitement important peut consister à filtrer certains mots. Par exemple, des mots outils ou "stop words" (mots fonctionnels tels que les prépositions, articles, etc.) peuvent être supprimés. Le but est alors de seulement conserver les mots ayant une signification représentative du texte.

3.2 Les N-grammes de mots

Il existe beaucoup de mots ayant la même forme, mais des sens différents. Par exemple, "prix" n'a pas le même sens dans "prix Goncourt", "grand prix" ou "prix marchandise". Ces mots augmentent l'ambiguïté du sens des textes (classification erronée). En utilisant les N-grammes de mots (N mots consécutif), un sens parmi d'autres est favorisé. Par exemple, "actes biologie" (issu des données d'ITESOFT) est un bigramme de mots particulièrement pertinent.

Nous donnons ci-dessous des exemples (issus des données d'ITESOFT) de N-grammes de mots :

- N=1 (unigramme) : "biologie", "médicale", "frais", "accessoires" et "maladie".
- N=2 (bigrammes) : "biologie médicale", "frais maladies" et "malade n^o ".
- N=3 (trigrammes) : "malade n^o facture", "prescription actes renseignements".

Paradis et Nie (2005) appliquent une telle représentation afin d'effectuer une classification de documents. La méthode consiste à classer les documents bruités en se fondant sur le filtrage

du contenu avec les N-grammes de mots et les entités nommées sur des documents de type "appels d'offres". Les travaux de Tan et al. (2002) utilisent des bigrammes ou des unigrammes de mots comme descripteurs pour la représentation des données pour une tâche de classification. Les résultats révèlent que l'utilisation des bigrammes sur ces données améliore les résultats de manière significative.

Notons que l'exemple donné précédement montre clairement que l'utilisation de N-grammes de mots favorise un sens parmi d'autres. Cependant, en se fondant sur le groupement de mots, nous pourrions dans certains cas dégrader la classification en introduisant une quantité supplémentaire de bruit. Par exemple dans le cas d'utilisation d'un trigramme de mots, les trois mots le composant vont être éloignés du sens de chacun des mots.

3.3 Les lemmes et stemmes

La lemmatisation consiste à représenter chaque mot par sa forme canonique. Ainsi, les verbes par leur forme infinitive et les noms par leur forme au singulier sont pris en compte. L'intérêt principal de la lemmatisation est la substitution des mots par leur racine ou leur lemme. Le processus permet une réduction du nombre de descripteurs. Par exemple, le remplacement des mots *conductrice*, *conducteur* par l'unique racine *conducteur* semble être avantageux tout comme le remplacement des formes conjuguées *franchit* et *franchi* par le lemme *franchir*.
La stemmatisation (radicalisation) consiste à supprimer tous les affixes d'un mot. Par affixes on entend : suffixe (défin-**ition**), préfixe (**sur**-consommation).

Ces techniques sont principalement utilisées pour réduire l'espace de représentation des documents et faire ressortir les traits similaires entre les mots. De nombreux travaux expérimentent ces techniques afin d'améliorer les performances de classification. Gonçalves et Quaresma (2005) appliquent sur un même corpus différents prétraitements (suppression de "stop words", suppression des "stop words" avec lemmatisation). Les travaux de Sjöblom (2002) appliquent différentes méthodes fondées sur les formes graphiques ou sur les lemmes d'un même corpus pour pouvoir comparer leurs apports dans l'analyse des données textuelles. Plus récemment, Lemaire (2008) révèle qu'il faut être prudent dans l'utilisation de la lemmatisation dans le cadre de l'extraction des significations à partir du contexte pour une tâche consistant à répondre automatiquement à un questionnaire. En effet, avec la lemmatisation nous risquons de perdre des informations cruciales car les contextes des mots en forme singulière et plurielle (par exemple, les mots "président" et "présidents") peuvent se révéler différents suggérant des concepts distincts.

Afin de lemmatiser les corpus propres à nos travaux, nous utilisons l'outil *TreeTagger*[2] (Schmid (1995)), développé par *l'Institute for Computational Linguistics de l'Université de Stuttgart*. Le *TreeTagger* est un système d'étiquetage automatique fondé sur un algorithme d'arbre de décision (Quinlan (1986)) pour effectuer l'analyse grammaticale.

3.4 Les N-grammes de caractères

Le N-gramme de caractères est une séquence de N caractères issus d'une chaîne de caractères. La notion de N-grammes de caractères a été utilisée de manière fréquente dans l'identification de la langue ou dans l'analyse de corpus oraux (Jalam et Teytaud (2001)). Dans les

[2]http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

recherches récentes, elle est utilisée pour l'acquisition et l'extraction des connaissances dans les corpus. De nombreux travaux (Cavnar et Trenkle (1994), Náther (2005)) utilisent les N-grammes de caractères comme méthode de représentation de documents d'un corpus pour la classification.

L'ensemble des N-grammes de caractères (en général, N varie de 2 à 5) est le résultat du déplacement d'une fenêtre de N cases sur le texte. Ce déplacement s'effectue par étapes, une étape correspondant à un caractère. Ensuite les fréquences des N-grammes de caractères sont calculées. Par exemple, nous avons les N-grammes ci-dessous (N=3) avec la phrase suivante : *"la nourrice nourrit le nourrisson"*

Trigrammes = [la_=1, a_n=1, _no=3, nou=3, our=3, urr=3, rri=3, ric=1, ice=1, _ce=1, e_n=2, rit=1, it_=1, t_l=1, _le=1, le_=1, ris=1, iss=1, sso=1, son=1].
Dans cet exemple, l'espace est représenté par le caractère "_", pour faciliter la lecture[3].

De nombreux travaux utilisent les N-grammes comme descripteurs de documents pour leur classification. (Junker et Hoch (1997)) présente une étude sur une représentation fondée sur les N-grammes de caractères pour évaluer la classification de textes issus d'OCR (Reconnaissance Optique de Caractères) et les textes correspondant en ASCII (non-OCR). Jalam et Chauchat (2002) expliquent les raisons pour lesquelles les N-grammes donnent des résultats intéréssants.

Les avantages que présentent les techniques qui s'appuient sur les N-grammes de caractères sont :

- Les N-grammes permettent de capturer automatiquement la racine des mots les plus fréquents. Il n'est pas nécessaire d'appliquer une étape de recherche de racine et/ou de lemmatisation.
- Ces descripteurs sont indépendants de la langue employée dans le corpus. Il n'est pas nécessaire d'utiliser des dictionnaires, ni de segmenter les documents en mots.
- Les N-grammes sont tolérants aux fautes d'orthographe et aux déformations causées lors de la reconnaissance de documents (système OCR). Lorsqu'un document est reconnu à l'aide du système OCR il y a souvent une part non négligeable de bruit. Par exemple, il est possible que le mot "feuille" soit lu "teuille". Un système fondé sur les N-grammes prendra en compte les autres n-grammes comme "eui", "uil", etc.

Cormack et al. (2007) appliquent les N-grammes de caractères pour la classification de petits documents tels que les polluriels (SPAM), courriers électroniques, SMS. D'autres travaux utilisent les N-grammes pour la classification de langues complexes. Dans (Mansur et al. (2006)) une tâche de classification d'un corpus d'actualités (newspaper) du langage *Bangali*[4] s'appuie sur la même technique que Cavnar et Trenkle (1994) et Náther (2005) pour représenter les documents du corpus. Les expérimentations fondées sur diverses valeurs de N (de 2 à 5-grammes) révèlent de bons résultats avec N = 3 et N = 4. Vardhan et al. (2007) montrent également de bonnes performances pour la classification de textes du langage *Telugu*[5] en se fondant sur les 3-grammes.

[3] L'espace entre les mots est considéré comme un caractère à part entière.
[4] Langue parlée au Bengale et au Bangladesh
[5] Langue parlée en Inde

3.5 Traitement numérique

Nous présentons dans cette section divers traitements possibles afin d'obtenir une représentation quantifiable d'un terme.

3.5.1 Représentation fréquentielle

Un traitement numérique simple et intuitif ("document frequency") consiste à calculer la fréquence des descripteurs dans chaque document. Une formule qui calcule le poids du mot t dans le document D est donné ci-dessous :

$\boxed{TF(t, D) = \text{fréquence du descripteur } t \text{ dans le document } D.}$

Un traitement numérique trivial supplémentaire peut consister à effectuer divers élagages selon la fréquence des descripteurs dans le corpus.
Un premier élagage consiste en la suppression des descripteurs dont la fréquence est en dessous d'un certain seuil (Roche et Kodratoff (2006)). L'idée sous-jacente est que ces descripteurs apportent peu d'informations représentatives. Elles peuvent d'ailleurs constituer du bruit (fautes d'orthographe, etc.).
Nous pouvons également supprimer les descripteurs "pauvres sémantiquement", tels que les mots généraux (*comme, pourquoi, chose, etc.*) et "stop-word". Notons que de tels mots sont la plupart du temps très fréquents.

Les travaux décrits ci-dessous proposent une mesure différente du TF afin d'attribuer un poids aux descripteurs.

3.5.2 Représentation avec TF.IDF

La représentation fréquentielle se fonde sur le nombre d'occurrences du descripteur dans le corpus. Cependant, en procédant ainsi nous donnons une importance trop grande aux descripteurs qui apparaissent très souvent dans toutes les classes et qui sont peu représentatifs d'une classe en particulier.

Nous trouvons dans la littérature (Salton et Buckley (1988)) une autre mesure de poids connue sous le nom TF.IDF (**T**erm **F**requency **I**nverse **D**ocument **F**requency). Elle permet de mesurer l'importance d'un mot en fonction de sa fréquence dans le document (TF = Term Frequency) pondérée par la fréquence d'apparition du terme dans tout le corpus (IDF = Inverse Document Frequency). Cette mesure permet de donner un poids plus important aux mots discriminants d'un document. Ainsi, un terme apparaissant dans tous les documents du corpus aura un poids faible.

Le poids d'un descripteur t_k dans un document D_j est calculé ainsi :

$$Tf.idf(t_k, D_j) = TF(t_k, D_j)IDF(t_k) \tag{1}$$

où $TF(t_k, D_j)$ est le nombre d'occurrences de t_k dans D_j, $IDF(t_R) = \frac{log|S|}{DF(t_R)}$ avec |S| le nombre de documents dans le corpus et $DF(t_R)$ est le nombre de documents contenant t_R.

D'autres variantes du TF.IDF qui n'ont pas été utilisées dans nos travaux sont données par (Nobata et al. (2003)) :

$$Tf.idf(t_k, D_j) = \frac{TF(t_k, D_j) - 1}{TF(t_k, D_j)} IDF(t_k) \quad (2)$$

$$Tf.idf(t_k, D_j) = \frac{TF(t_k, D_j)}{TF(t_k, D_j) + 1} IDF(t_k) \quad (3)$$

À la différence de la formule (1) qui utilise la fréquence des termes à l'état brut (mesure utilisée dans notre étude), les deux formules (2) et (3) sont utilisées pour normaliser la fréquence. Ainsi, les travaux de Robertson et Walker (1994) rapportent que la méthode (3) est plus efficace dans la recherche documentaire.

Une des particularités du TF.IDF tient au fait que cette mesure donne un poids faible aux mots présents dans l'ensemble des documents car ils ne sont pas discriminants (cas des "stop words"). Un tel traitement peut donc se révéler particulièrement pertinent pour le processus fondé sur les N-grammes de caractères qui n'utilisent pas de prétraitements (comme la suppression des "stop words").

Un autre traitement permettant de filtrer les descripteurs peut s'appuyer sur des traitements linguistiques tels que présentés dans la section suivante.

3.6 Traitement linguistique : Utilisation de Connaissances morpho-syntaxiques

La plupart des travaux précédemment décrits consistent à supprimer les descripteurs non pertinents. Nous pouvons également nous focaliser sur la détermination des types de descripteurs les plus représentatifs pour chaque document en sélectionnant les mots respectant une ou plusieurs catégories grammaticales adaptées (nom, verbe, adjectif, etc.).

La sélection d'attributs selon les catégories grammaticales permet d'identifier par exemple des traits de jugement subjectif pour la classification de documents par genre ou par opinion comme dans les travaux de Genereux et Santini (2007) (critiques de films, débats politiques, etc.). Dans ce cas, les adjectifs voire les adverbes sont bien adaptés (Benamara et al. (2007)). Par ailleurs, nous trouvons dans les approches de Greevy et Smeaton (2004) une classification de documents racistes issus du web. Cette étude montre que les documents racistes contiennent beaucoup plus d'adjectifs que les documents non racistes. Les nombreux travaux de la littérature utilisant ces descripteurs nous ont conduit à les expérimenter. Nous proposons dans ces travaux de mesurer l'impact de l'utilisation de descripteurs fondés sur la sélection de catégorie(s) grammaticale(s), sur des données de type rétro-conversion de documents par lecture OCR. L'utilisation de ces descripteurs peuvent permettre de réduire de manière conséquente la taille du corpus résultant, et ainsi permettre de réduire le temps de traitement nécessaire ainsi que l'espace mémoire. Après avoir donné une description de différentes méthodes de représentations de texte, nous présenterons dans la prochaine section notre approche : HYBRED.

4 L'approche HYBRED

L'approche HYBRED (***HYB****rid* ***RE****presentation of* ***D****ocuments*) propose d'utiliser la qualité des meilleurs descripteurs décrits précédemment afin de les combiner. Dans un premier temps, nous présentons les motivations ayant conduit à cette proposition.

4.1 Motivation

Pour répondre à notre problématique, il est indispensable de proposer une approche qui soit pertinente et qui améliore les performances de la classification de données bruitées.

Les différentes expérimentations que nous avons menées sur les corpus d'ITESOFT s'appuient sur les descripteurs précédemment explicités : le mot, le n-gramme de mots, le n-gramme de caractères et la selection de catégories grammaticales ainsi que l'utilisation du *TF* ou du *Tf-Idf*. Ces travaux nous ont permis d'identifier les descripteurs les plus efficaces pour notre tâche de classification. Nous proposons alors l'approche HYBRED qui combine les descripteurs les plus pertinents.

4.2 Quelles approches combiner ?

L'objectif de cette section est de motiver le choix des descripteurs qui pourront être combinés.

4.2.1 Choix des approches

Les expérimentations qui seront présentées dans la section 5 ainsi que les résultats issus des travaux de la littérature nous ont amené à sélectionner trois méthodes :

- Étiquetage grammatical.
- N-grammes de caractères.
- Filtrage statistique.

Le choix s'est porté sur ces trois descripteurs pour les raisons suivantes. L'application de l'étiquetage grammatical a pour but de sélectionner les données respectant une catégorie ou un groupe de catégories grammaticales données (nom, verbe, adjectif, nom-verbe, nom-adjectif, etc.). L'objectif principal de ce traitement est de ne conserver que les données ayant une information sémantique pertinente pour une tâche de classification. De nombreux travaux de la littérature comme ceux de Kohomban et Lee (2007) montrent par exemple que les noms sont très porteurs de sens. Ces résultats, qui permettent d'améliorer la tâche de classification, confirment ceux que nous avions obtenus lors de précédents travaux (Bayoudh et al. (2008)).

La représentation des données selon les N-grammes de caractères est motivée par la complexité des données que nous manipulons (données bruitées issues de la rétroconversion d'OCR). En effet, l'utilisation des N-grammes de caractères est adaptée dans le cadre des données issues d'OCR (Junker et Hoch (1997)).

Le dernier pocessus proposé consiste à appliquer une mesure statistique afin d'attribuer un poids aux descripteurs. En leur attribuant un poids, nous favorisons les plus discriminants pour une classe particulière.

4.2.2 Ordre des approches

La sélection des descripteurs est plus ou moins adaptée et dépend de l'ordre dans lequel ces approches vont être utilisées.

L'association d'étiquettes grammaticales peut seulement s'effectuer à partir des mots "au complet". En d'autres termes, elle n'est pas applicable sur les mots tronqués par les N-grammes de caractères. En effet, utiliser des catégories grammaticales après l'extraction de n-grammes

de mots serait incohérent linguistiquement et impossible avec des n-grammes de caractères, les mots n'étant plus identifiables. L'ordre établi consiste donc à appliquer un filtrage grammatical (sélection des mots selon leur étiquette) suivi par une représentation des N-grammes. Le fait de finir le traitement par un filtrage statistique se justifie par la représentation de chaque document par un vecteur de K éléments (les éléments représentent les K N-grammes). Ces éléments ne sont pas tous discriminants, en leur attribuant un poids nous favorisons les plus significatifs pour caractériser une classe.

4.3 Comment combiner les approches ?

Dans la section précédente, nous avons établi un ordre dans l'application des différents traitements. Dans cette section, nous allons détailler la manière de combiner nos approches.

Dans un premier temps, la sélection des mots avec des étiquettes grammaticales est effectuée. La sélection selon les étiquettes (Nom-Verbe) sur la phrase "`or le bijoux plaqué or a du charme.`", nous donne le résultat suivant : "`bijoux plaqué or a charme`". Notons que le filtrage grammatical permet de distinguer le mot "or" de type *conjonction de coordination* comparativement au *nom* "or".

Après ce premier traitement, nous représentons les mots extraits par les N-grammes de caractères. L'application de la représentation N-grammes de caractères nous donne trois possibilités de représentation :

- La première représentation peut être considérée comme un sac de mots sélectionnés grammaticalement. L'application des N-grammes avec N=5 nous donne par exemple le résultat suivant :

  "_bijo, bijou, ijoux, joux_, oux_p, ux_pl, x_pla, _plaq, plaqu, laqué, aqué_, qué_o, ué_or, é_or_, _or_a, or_a_, **r_a_c**, **_a_ch**, **a_cha**, _cham, chamr, harme, arme_"

  Cette application est erronée car elle rajoute du bruit et des N-grammes inutiles, par exemple `a_cha` est un des N-grammes qui représente du bruit (N-gramme issu du fragment "a du charme" pour lequel le mot "du" a été supprimé). En effet, le fait d'éliminer des mots de la phrase initiale entraîne la construction de suites de mots non pertinents (et donc des N-grammes incorrects).

- Un deuxième type de représentation consiste à appliquer des N-grammes de caractères pour chacun des mots extraits séparément. Nous aurons comme résultat :

  "_bijo, bijou, ijoux, joux_, _plaq, plaqu, laqué, aqué_, _cham, chamr, harme, arme_"

  Cette représentation corrige les défauts causés par la précédente méthode. Elle n'introduit pas de bruit mais elle souffre de perte d'information notamment sur les mots courts. Par exemple, en appliquant les N-grammes de caractères avec $N \geq 5$ le nom "or" ne peut être identifié. Cette suppression occasionne une perte d'information.

- Les deux premières représentations ont donc des défauts majeurs liés à l'introduction de **bruit** (première méthode) et du **silence** (deuxième méthode). Pour cela nous avons introduit un **principe de frontière**. Celui-ci permet de remplacer les mots supprimés par une frontière. Cette méthode corrige le défaut de rajout du bruit causé lors de la première représentation. Il permet également de prendre en considération des groupes

de mots (par exemple, "plaqué or"). Le résultat obtenu selon le principe de frontière est montré ci-dessous :

"`X bijoux plaqué or a X charme`", le "X" représente la frontière.

L'application de la méthode des 5-grammes donne le résultat ci-dessous :

"`_bijo, bijou, ijoux, joux_, oux_p, ux_pl, x_pla, _plaq, plaqu, laqué, aqué_, qué_o, ué_or, é_or_, _or_a, or_a_, _cham, chamr, harme, arme_ `"

Après avoir présenté les descripteurs les plus pertinents pour notre tâche, avoir défini l'ordre d'application des descripteurs et enfin après avoir discuté de la manière de combiner au mieux les descripteurs, nous présentons dans la section suivante notre approche appelée HYBRED.

4.4 Approche HYBRED

Dans cette section, nous présentons le principe que nous avons retenu pour notre système de représentation des données. Le principe général est résumé dans l'algorithme ci-dessous qui sera détaillé dans cette section.

Entrées : L'ensemble des textes constituant le corpus.
Sorties : Matrice.
forall *Documents* **do**
  Extraction des mots selon une étiquette grammaticale (a)
  Application du principe de frontière (b)
  Représentation des mots extraits selon les N-grammes de caractères (c)
  Attribution de poids selon la mesure TF.IDF (d)
end

Algorithme 1 : HYBRED

4.4.1 Description d'HYBRED

Étape (a) : sélection selon une étiquette grammaticale

Une sélection des données selon une étiquette grammaticale propose de ne sélectionner que les termes appartenant à une ou plusieurs catégories grammaticales données, comme les *noms* et les *verbes*.

Étape (b) : application du principe de frontière

Dans les travaux de Bourigault (1994), nous trouvons une application du principe de frontière. LEXTER, développé par D. Bourigault est un outil d'extraction de la terminologie. Il effectue une extraction de groupes nominaux (syntagmes nominaux) par repérage des marqueurs de frontières. Ces frontières sont déterminées linguistiquement (exemples de frontière : "préposition + adjectif possessif", "préposition + article indéfini", etc.). Les candidats termes, à savoir les groupes nominaux maximaux, sont extraits sur la base de leur position relative aux frontières.

Dans notre étude, les mots apportant peu d'informations (avec des étiquettes grammaticales moins pertinentes) sont remplacés par une frontière. L'objectif reste le même que dans

LEXTER. En effet, nous prenons en considération les groupes de mots pertinents situés entre les frontières. Cependant, la différence tient au fait que nos frontières sont les mots ayant des étiquettes grammaticales moins pertinentes pour les tâches de classifications (adverbe, préposition, etc.) et ne s'appuient pas sur des règles linguistiques comme dans LEXTER.

Étape (c) : représentation avec les N-grammes
Après avoir conservé les données appartenant à une étiquette grammaticale et appliqué le principe de frontière, vient l'étape de représentation avec les N-grammes de caractères. Il s'agit d'une fusion des N-grammes des différents fragments séparés par la frontière.

$$\texttt{Nbr-N-grammes} = \sum_{i \in \{ensemble\ \ des\ \ fragments\}} \texttt{N-grammes}(fragment_i)$$

Après avoir effectué la représentation avec les N-grammes de caractères, nous réalisons une étape de filtrage de N-grammes non pertinents. Cette étape consiste à supprimer les N-grammes peu fréquents et qui peuvent constituer du bruit (N-grammes < seuil (fixé manuellement à 30)).

Étape (d) : filtrage statistique
Enfin, de manière similaire aux très nombreux travaux de la littérature, nous avons appliqué un filtre statistique fondé sur le TF.IDF afin de ne conserver que les descripteurs discriminants. Le principe du Tf-Idf appliqué ici est décrit dans la section 3.5.2.

4.4.2 Exemple de l'application d'HYBRED

Cette section développe un exemple complet de l'approche HYBRED. Pour ce faire, nous considérons la phrase "`Il faut une infinie patience pour attendre toujours ce qui n'arrive jamais`".

(a) La sélection des données selon la combinaison NVA (Nom Verbe Adjectif) donnera comme résultat : "`faut infinie patience attendre arrive`".

(b) L'application du principe de frontière, nous donne :
"`X faut X infinie patience X attendre X arrive X`".

(c) La représentation sous N-grammes ou N=3 aura comme résultat :

Mot	N-grammes de caractères
[_faut_]	[_fa, fau, aut, ut_]
[_infinie patience_]	[_in, inf, nfi, fin, ini, nie, ie_, e_p, _pa, pat, ati, tie, ien, enc, nce, ce_]
[_attendre_]	[_at, att, tte, ten, end, ndr, dre, re_]
[_arrive_]	[_ar, arr, rri, riv, ive, ve_]

Ainsi, nous pouvons calculer la somme de tous les 3-grammes :
N-grammes("_faut_") + N-grammes("_infinie patience_") + N-grammes("_attendre_")
+ N-grammes("_arrive_").

Nous obtenons :
{_fa, fau, aut, ut_, _in, inf, nfi, fin, ini, nie, ie_, e_p, _pa, pat, ati, tie, ien, enc, nce, ce_, _at, att, tte, ten, end, ndr, dre, re_, _ar, arr, rri, riv, ive, ve_}

Enfin, les filtrages numériques (élagage selon un seuil et TF.IDF) seront appliqués.

5 Expérimentations

Dans cette section, nous présentons les différentes expérimentations que nous avons réalisées pour déterminer les descripteurs pertinents et tester la pertinence de notre approche. Tout d'abord, nous présentons le protocole expérimental sur lequel nous nous appuyons puis les résultats obtenus lors de la classification.

5.1 Protocole expérimental

Pour évaluer la pertinence des différents descripteurs, nous avons choisi d'utiliser les algorithmes de classification à apprentissage supervisé suivants : les K plus proches voisins (K-PPV), un classificateur fondé sur les machines à support vectoriel (SVM) et un algorithme probabiliste (Naïve Bayes). Le choix s'est porté sur ces derniers parce qu'il s'agit des algorithmes d'apprentissage automatique qui sont souvent les plus performants pour la classification automatique de textes (Sebastiani (1999), Yang et Liu (1999)).

Notons que nous n'avons pas implémenté ces algorithmes, mais nous avons utilisé le logiciel "Weka[6]" (Witten et al. (1999)). Weka contient un ensemble de classes et d'algorithmes en Java implémentant les principaux algorithmes de "fouille de données". Il contient des outils pour le prétraitement de données, la classification, la régression, le groupement et la visualisation. Les paramètres que nous avons utilisés avec les algorithmes de classification sont ceux définis par défaut dans Weka (K-PPV avec différentes valeurs de K testées : la valeur K=1 a été retenue car elle donne des résultats satisfaisants à partir de nos données, SVM avec l'algorithme multi-classes SMO).

Afin de déterminer quels descripteurs sont les plus pertinents, nous comparons les résultats obtenus avec les trois algorithmes. Les résultats de la classification sont présentés selon la précision (*accuracy*) des algorithmes (classement correct des documents dans les catégories adaptées).

$$Précision = \frac{nombre\ de\ documents\ correctement\ classés}{ensemble\ des\ documents\ classés}$$

Une précision de 100% signifie que tous les documents sont classés dans la catégorie pertinente.
Par ailleurs, cette mesure de précision a été calculée après l'application d'une 10-validation croisée (principe détaillé dans la section 2.2.1)

Nous rappelons que les documents sont représentés selon une approche vectorielle. Nous effectuons une suppression des informations rajoutées par OCR comme les coordonnées de chaque mot dans les documents. En outre, nous avons expérimenté le calcul de fréquence de chaque descripteur dans le corpus ainsi que la pondération avec la mesure Tf.Idf.

Pour évaluer les performances de l'utilisation de nos différents descripteurs pour une tâche de classification, nous avons utilisé trois corpus de test :

[6] www.cs.waikato.ac.nz/ml/weka

Corpus A
Le premier jeu d'essai comporte 250 fichiers en provenance d'ITESOFT se répartissant en 18 catégories qui représentent des frais, des factures, des attestations (transport, hospitalière, médecin, etc). Ce corpus en français liste des fichiers images qui peuvent être :

- Des images sur des documents manuscrits.
- Des formulaires.
- Des documents imprimés ou dactylographiés en fichiers de texte.

Un fichier XML est associé à chaque image (le fichier XML est le résultat OCR de rétroconversion de l'image). Le fichier contient une description "structurelle" du document organisé en blocs, lignes, mots, caractères.
Les textes contiennent peu de phrases bien formulées en langage naturel. Notons que certaines classes contiennent plus de documents que d'autres (distribution non équilibrée entre les classes). La grande complexité des données est due au faible contenu des documents, mais surtout à un ensemble d'apprentissage de faible taille.

Corpus B
Le deuxième corpus utilisé pour les expérimentations provient aussi d'ITESOFT. Il comporte 2000 fichiers répartis dans 24 catégories. Les documents en français sont issus d'une rétroconversion d'OCR.
Les catégories représentent des bulletins, des certificats, des avis d'impôt, etc. La principale caractéristique du corpus est la diversité des documents et le faible contenu de ces derniers. Mais la grande difficulté est la distribution des documents. Nous avons des catégories qui ne contiennent que 1 à 5 documents en comparaison avec d'autres qui contiennent plus de 250 documents (répartition très hétérogène).

Corpus C
Le dernier corpus est une collection de dépêches (en français) obtenue manuellement à partir de sites d'actualité sur internet. Il comporte 65 documents répartis en 5 catégories. L'idée de l'utilisation de ce dernier est de pouvoir comparer les performances du système avec un corpus plus riche et moins bruité.
Les catégories sont des articles de presse sur le Président Sarkozy traitant du thème de vie privée, la diffusion de vidéo de TF1 sur les sites Daylimotion et Youtube, les élections à la ville de Paris, du sport (ski) et la dernière classe représente les actions de l'association "des enfants Don Quichotte" sur le problème des personnes sans domicile fixe. Les documents contiennent peu de bruit et sont d'une taille conséquente pour l'apprentissage.

Le tableau 1 résume les principales caractéristiques des trois corpus exploités.

5.2 Résultats expérimentaux

Dans cette section, nous allons présenter les expérimentations selon les différents descripteurs et l'approche HYBRED. Notons que nous avons appliqué un filtrage des N-grammes peu fréquents, en fixant un seuil à 30. En dessous de ce seuil, les N-grammes ne sont pas pris en compte. Nous donnerons les résultats sur le corpus B, pour les autres corpus les expérimentations sont données en Annexe. En effet, ce corpus B est le plus représentatif en termes de taille et de difficulté de traitement (données hétérogènes et bruitées). En outre, une des difficultés du

	Corpus A	Corpus B	Corpus C
Nombre de documents	250	2000	65
Nombre de catégories	18	24	5
Taille (en Mo)	0.434	2.12	0.211
Nombre de mots	73855	260752	31711
Type de textes	rétro-conversion OCR	rétro-conversion OCR	Articles journaux

TAB. 1 – *Résumé des principales caractéristiques des trois corpus.*

traitement du corpus B tient au fait qu'en moyenne, chaque document est moins riche en terme de nombre de mots (130 mots par document pour le corpus B) comparativement aux corpus A (295 mots par document) et C (488 mots par document).

Le tableau 2 présente les résultats obtenus avec les différents descripteurs en appliquant une validation croisée (10-CV).

Algorithmes	K-PPV		SVM		NB	
mesure	Fréquentiel	TF.IDF	Fréquentiel	TF.IDF	Fréquentiel	TF.IDF
Mot	91.1	**91.1**	**95.8**	**95.8**	**94.1**	**93.8**
2-mots	**92.2**	90.9	93.7	93.7	91.9	92.2
3-mots	90.5	90.5	90.1	89.9	82.8	86.1
2-caractères	73.7	72.6	89.6	88.2	74.3	58.5
3-caractères	85.7	86.0	**96.5**	**96.8**	**93.4**	**91.9**
4-caractères	**95.0**	**96.1**	96.0	96.3	93.1	90.7
5-caractères	91.4	92.5	96.2	95.6	92.0	90.8
Lemme	92.3	93.8	95.4	**95.5**	93.7	94.4
N	91.1	**93.0**	95.6	95.1	93.6	94.6
V	88.2	87.5	88.4	87.8	85.2	84.9
NV	92.4	92.7	95.5	95.5	**94.1**	94.3
NVA	**93.3**	92.6	**95.6**	**95.8**	**94.1**	94.5
NA	92.8	92.4	**95.6**	95.4	93.9	**94.8**
VA	92.0	91.4	93.7	93.7	91.7	91.4

TAB. 2 – *Résultats du corpus B avec les différents descripteurs (précision).*

Nous observons que les meilleurs résultats sont obtenus avec l'algorithme SVM par rapport à K-PPV et à Naïve Bayes. La représentation avec les N-grammes de caractères se comporte bien avec les trois corpus. En général, nous remarquons que les résultats se détériorent significativement quand N=2. L'application de l'analyse morphosyntaxique ou la sélection selon une étiquette grammaticale peut, dans certains cas, se révéler efficace dans son utilisation et sa capacité à ne sélectionner que les données discriminantes (ce qui réduit significativement l'espace de représentation comme nous allons le montrer dans la section 5.3). Nos résultats montrent qu'une combinaison selon les NV (**N**om**V**erbe), NVA (**N**om**V**erbe**A**djectif) et NA

(**N**om**A**djectif) est la plus pertinente parmi les combinaisons possibles.

Dans le corpus A, les résultats (présentés en Annexe) sont en général meilleurs que les résultats obtenus avec le corpus B. Ceci est dû à la complexité du corpus B (corpus bruité possédant en moyenne moins de mots par document). Enfin, avec le corpus C, nous obtenons les meilleurs résultats. Cela s'explique par le corpus en lui-même (textes journalistiques), qui est peu bruité et chaque document est assez riche en terme de nombre de mots.

Le tableau 3 présente les résultats obtenus en appliquant l'approche HYBRED avec le corpus B. Les résultats des corpus A et C sont donnés en Annexe 3.

Algorithme K-PPV (TF.IDF)						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	74.8	85.5	74.6	72.7	74.6	77.3
3-caractères	85.0	85.5	95.8	87.0	86.3	86.6
4-caractères	85.0	86.5	**96.7**	92.6	92.1	**90.2**
5-caractères	**91.8**	**88.4**	93.0	**93.4**	**92.4**	90.0
Algorithme SVM (TF.IDF)						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	89.5	89.9	87.4	86.43	89.0	88.1
3-caractères	96.4	**94.2**	96.6	**96.9**	**96.8**	**96.3**
4-caractères	**96.5**	93.8	**98.0**	96.8	96.7	95.8
5-caractères	96.4	93.2	96.8	96.8	96.7	95.2
Algorithme NB (TF.IDF)						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	61.4	**92.6**	60.7	59.4	73.5	63.6
3-caractères	61.4	88.3	60.7	**92.2**	73.5	91.4
4-caractères	**92.6**	88.3	**96.9**	**92.2**	**92.7**	**91.9**
5-caractères	**92.6**	86.9	93.1	92.1	92.4	91.5

TAB. 3 – *Précision obtenue avec l'approche HYBRED pour le corpus B*

Dans tous les cas, nous obtenons les meilleures performances avec l'algorithme **SVM** par rapport aux K-PPV et Naïve Bayes. Nous remarquons que la sélection NV (Nom Verbe) associée aux 4-grammes donne des résultats très satisfaisants sur le corpus B. Nous remarquons globalement ces mêmes comportements sur les corpus A et C.

5.3 Synthèse

Comme nous avons pu le constater lors des expérimentations, la méthode HYBRED fondée sur la combinaison des descripteurs a tendance à améliorer les performances des classificateurs par rapport à l'utilisation de chaque descripteur séparément. Nous pouvons observer cette amélioration sur les trois corpus, notamment sur le corpus B qui est le plus complexe à classifier. Le tableau 4 présente une comparaison entre l'approche proposée avec une combinaison NV

(**N**om**V**erbe) associée à une représentation 4-grammes de caractères et les différents descripteurs.

	Corpus A			Corpus B			Corpus C		
algorithmes	K-PPV	SVM	NB	K-PPV	SVM	NB	K-PPV	SVM	NB
mot	96.5	97.5	**96.7**	91.1	95.8	93.8	95.3	98.4	**98.4**
3-caractères	94.7	97.9	93.5	86.0	96.8	91.9	98.4	**100**	82.8
4-caractères	**97.5**	98.3	94.3	96.1	96.3	90.7	98.4	**100**	82.8
5-caractères	96.7	98.3	95.1	92.5	95.6	90.8	**100**	**100**	85.9
NV	95.9	98.0	**96.7**	92.7	95.5	94.3	96.4	98.0	87.5
NVA	95.5	98.0	**96.7**	92.6	95.8	94.5	96.8	98.2	79.6
NA	95.1	98.0	**96.7**	92.4	95.4	94.8	98.0	98.0	85.9
HYBRED	96.8	**98.4**	93.6	**96.7**	**98.0**	**96.9**	**100**	**100**	79.6

TAB. 4 – *Tableau de comparaison entre HYBRED et des méthodes "classiques"*

En général, des améliorations de la précision ont été obtenues en appliquant l'approche HYBRED. Ainsi, le tableau 4 montre qu'HYBRED améliore toujours les résultats (de manière plus ou moins significative selon les corpus) avec l'algorithme SVM. Ceci est particulièrement intéressant, car cet algorithme est celui qui a le meilleur comportement, ce que nos expérimentations ont confirmé à partir des trois corpus. Par ailleurs, cette amélioration est particulièrement importante avec le corpus le plus représentatif et le plus complexe (corpus B).

Dans le tableau 5, nous présentons une comparaison de l'espace de représentation (taille) avec et sans appliquer l'approche HYBRED. Nous remarquons que l'application d'HYBRED réduit de manière significative l'espace de représentation. Les résultats obtenus selon l'approche HYBRED sont donnés avec la combinaison NV (Nom Verbe) suivie d'une représentation 4-grammes de caractères.

Espace de représentation	Sans application d'HYBRED		Après application d'HYBRED
	Mot	N-grammes (N=4)	NV + N-grammes (N=4)
Corpus A	12307	2087	1603
Corpus B	37837	4485	3294
Corpus C	5417	1274	876

TAB. 5 – *Tableau de comparaison de l'espace de représentation avec et sans HYBRED*

La taille de l'espace de représentation avec l'ensemble des combinaisons possibles avec HYBRED est détaillée en Annexe 2.

6 Conclusion

Dans cet article, nous avons proposé une nouvelle approche de classification automatique de documents textuels. Les expérimentations menées sur des jeux de données issus de la retro-conversion d'OCR produisent de bonnes performances de classification. Les perspectives à ce travail sont nombreuses. Tout d'abord, même si nous ne l'avons pas expérimenté dans nos travaux, l'application de connaissances sémantiques (par exemple, l'utilisation de dictionnaires spécialisés pour enrichir les descripteurs linguistiques sélectionnés) pourrait améliorer les performances. Cependant, de tels dictionnaires ne sont pas disponibles pour tous les domaines. L'autre perspective envisagée est liée aux techniques de classification, en d'autres termes, nous souhaitons expérimenter d'autres types d'algorithmes. Nous souhaitons également appliquer l'approche à d'autres types de corpus bruités afin de conforter la pertinence de notre approche dans un tel contexte. Nous pouvons enfin appliquer notre approche pour une tâche de classification de données d'opinion (en privilégiant les descripteurs linguistiques de type adjectif ou adverbe par exemple). En effet, de telles données d'opinion souvent issues de Blogs peuvent se révéler particulièrement bruitées rendant le traitement automatique complexe.

Références

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Soc. of London 53*, 370–418. reprinted in Biometrika 45(3/4) 293-315 Dec 1958.

Bayoudh, I., N. Béchet, et M. Roche (2008). Blog classification : Adding linguistic knowledge to improve the k-nn algorithm. In *Intelligent Information Processing*, pp. 68–77.

Benamara, F., C. Cesarano, A. Picariello, D. Reforgiato, et V. Subrahmanian (2007). Sentiment analysis : Adjectives and adverbs are better than adjectives alone. In *IADIS Applied Computing, Boulder, Colorado, U.S.A, 26/03/07-28/03/07*, pp. 203–206. ACM.

Bourigault, D. (1994). *LEXTER un Logiciel d'EXtraction de TERminologie. Application à l'extraction des connaissances à partir de textes.* Thèse en mathématiques, informatique appliquée aux sciences de l'homme, École des hautes Études en sciences sociales, Paris.

Cavnar, W. et J. Trenkle (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, pp. 161–175.

Cormack, G., J. Hidalgo, et E. Sánz (2007). Spam filtering for short messages. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, New York, NY, USA, pp. 313–320. ACM.

Genereux, M. et M. Santini (2007). Defi: classification de textes francais subjectifs. In *In: 3eme DEfi fouille de textes*, Grenoble, Switzerland.

Gonçalves, T. et P. Quaresma (2005). *Evaluating preprocessing techniques in a Text Classification problem*. São Leopoldo, RS, Brasil: SBC - Sociedade Brasileira de Computação.

Greevy, E. et A. F. Smeaton (2004). Classifying racist texts using a support vector machine. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on*

Research and development in information retrieval, New York, NY, USA, pp. 468–469. ACM.

Jalam, R. et J. Chauchat (2002). Pourquoi les n-grammes permettent de classer des textes ? recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques. In *6th International Conference on Textual Data Statistical Analysis, France*, pp. 381–390. IRISA-INRIA.

Jalam, R. et O. Teytaud (2001). Identification de la langue et catégorisation de textes basées sur les n-grammes. In H. Briand et F. Guillet (Eds.), *EGC*, Volume 1 of *Extraction des Connaissances et Apprentissage*, pp. 227–238. Hermes Science Publications.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In C. Nédellec et C. Rouveirol (Eds.), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Number 1398, Chemnitz, DE, pp. 137–142. Springer Verlag, Heidelberg, DE.

Junker, M. et R. Hoch (1997). Evaluating ocr and non-ocr text representations for learning document classifiers. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, Washington, DC, USA, pp. 1060–1066. IEEE Computer Society.

Kohomban, U. S. et W. S. Lee (2007). Optimizing classifier performance in word sense disambiguation by redefining sense classes. In *IJCAI*, pp. 1635–1640.

Lemaire, B. (2008). Limites de la lemmatisation pour l'extraction de significations. In *: S. Heiden and B. Peiden (eds.): 9th International Conference on the Statistical Analysis of Textual Data, JADT'2008*, Volume 2, Lyon, France, pp. 725–732.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. pp. 4–15. Springer Verlag.

Mansur, M., N. UzZaman, et M. Khan (2006). Analysis of n-gram based text categorization for bangla in a newspaper corpus. In *Proc. of 9th International Conference on Computer and Information Technology (ICCIT 2006)*, Dhaka, Bangladesh.

Memmi, D. (2000). Le modèle vectoriel pour le traitement de documents. Cahiers Leibniz 2000-14, INPG.

Nobata, C., S. Sekine, et H. Isahara (2003). Evaluation of features for sentence extraction on different types of corpora. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, Morristown, NJ, USA, pp. 29–36. Association for Computational Linguistics.

Náther, P. (2005). N-gram based text categorization, institute of informatics, comenius university.

Paradis, F. et J. Nie (2005). Filtering contents with bigrams and named entities to improve text classification. In *AIRS*, pp. 135–146.

Pisetta, V., H. Hacid, F. Bellal, et G. Ritschard (2006). Traitement automatique de textes juridiques. In R. Lehn, M. Harzallah, N. Aussenac-Gilles, et J. Charlet (Eds.), *Semaine de la Connaissance (SdC 06), Nantes*.

Platt, J. C. (1999). *Fast training of support vector machines using sequential minimal optimization*, pp. 185–208. Cambridge, MA, USA: MIT Press.

Pouliquen, B., D. Delamarre, et P. L. Beux (2002). Indexation de textes médicaux par extrac-

tion de concepts, et ses utilisations. In *: A. Morin and P. Sébillot (eds.): 6th International Conference on the Statistical Analysis of Textual Data, JADT'2002*, Volume 2, St. Malo, France, pp. 617–627.

Quinlan, J. (1986). Induction of decision trees. *Mach. Learn. 1*(1), 81–106.

Robertson, S. et S. Walker (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 232–241. Springer-Verlag New York, Inc.

Roche, M. et Y. Kodratoff (2006). Choix du taux d'élagage pour l'extraction de la terminologie. une approche fondée sur les courbes roc. *Revue RNTI (Revue des Nouvelles Technologies de l'Information) numéro spécial conférence EGC'06 E6*, 205–216.

Salton, G. et C. Buckley (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage. 24*(5), 513–523.

Salton, G., A. Wong, et C. S. Yang (1975). A vector space model for automatic indexing. *Commun. ACM 18*(11), 613–620.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. *In Proceedings of the ACL SIGDAT-Workshop, Dublin*.

Sebastiani, F. (1999). A tutorial on automated text categorisation. In A. Amandi et R. Zunino (Eds.), *Proceedings of the 1st Argentinian Symposium on Artificial Intelligence (ASAI'99)*, Buenos Aires, AR, pp. 7–35.

Sjöblom, M. (2002). Le choix de la lemmatisation. différentes méthodes appliquées à un même corpus. In *JADT : 6es Journées internationales d'Analyse statistique des Données Textuelles*.

Tan, C., Y. Wang, et C. Lee (2002). The use of bigrams to enhance text categorization. *Inf. Process. Manage. 38*(4), 529–546.

Vardhan, B. V., L. P. Reddy, et A. VinayBabu (2007). Text categorization using trigram technique for telugu script. *Journal of Theoretical and Applied Information Technology 3*(1-2).

Vinot, R., N. Grabar, et M. Valette (2003). Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'internet. In *actes du colloque TALN 2003, 11-14 juin 2003, Batz sur Mer, pages=257–284*.

Witten, I., E. Frank, L. Trigg, M. Hall, G. Holmes, et S. Cunningham (1999). Weka: Practical machine learning tools and techniques with java implementations. In: Proc ICONIP/ANZIIS/ANNES'99 Int. Workshop: Emerging Knowledge Engineering and Connectionist-Based Info. Systems. 192-196.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval 1*(1/2), 69–90.

Yang, Y. et X. Liu (1999). A re-examination of text categorization methods. In *SIGIR*, pp. 42–49. ACM.

Annexe 1

Nous donnons les résultats comparatifs des différents descripteurs obtenus sur les corpus A (tableau 6) et C (tableau 7).

Algorithmes	K-PPV		SVM		NB	
mesure	Fréquentiel	TF.IDF	Fréquentiel	TF.IDF	Fréquentiel	TF.IDF
mot	**95.7**	**96.5**	**97.9**	**97.5**	**96.7**	**96.7**
2-mots	88.7	85.5	90.3	86.7	91.9	89.5
3-mots	77.1	76.3	74.2	74.6	77.1	73.0
2-caractères	94.3	77.9	95.9	85.9	87.9	77.9
3-caractères	**96.3**	94.7	**97.9**	97.9	96.3	93.5
4-caractères	94.3	**97.5**	**97.9**	**98.3**	96.3	94.3
5-caractères	95.5	96.7	97.5	**98.3**	**96.7**	**95.1**
Lemme	95.3	95.1	95.8	**96.7**	95.1	95.9
N	95.9	95.9	97.1	97.1	96.7	97.1
V	84.7	83.5	86.3	83.9	92.7	84.3
NV	**96.3**	**95.9**	97.1	**98.0**	96.7	**96.7**
NVA	95.9	**95.9**	**97.5**	**98.0**	**97.1**	**96.7**
NA	95.5	95.1	**97.5**	**98.0**	**97.1**	**96.7**
VA	93.1	92	95.5	95.0	95.1	91.0

TAB. 6 – *Résultats du corpus A avec les différents descripteurs (précision).*

Algorithmes	K-PPV		SVM		NB	
mesure	Fréquentiel	TF.IDF	Fréquentiel	TF.IDF	Fréquentiel	TF.IDF
Mot	**96.8**	**95.3**	**98.4**	**98.4**	93.7	**98.4**
2-mots	90.6	89.0	96.8	93.7	**98.4**	93.7
3-mots	84.3	84.3	79.6	79.6	85.9	86.0
2-caractères	84.3	84.3	79.6	79.8	85.9	**85.9**
3-caractères	96.8	98.4	**100**	**100**	93.7	82.8
4-caractères	**98.4**	98.4	**100**	**100**	**95.3**	82.8
5-caractères	**98.4**	**100**	**100**	**100**	90.6	**85.9**
Lemme	90.6	90.6	98.2	**98.4**	92.1	95.3
N	91.1	**93.0**	**95.6**	95.1	93.6	94.6
V	88.2	87.5	88.4	87.8	85.2	84.9
NV	92.4	92.7	95.5	95.5	**94.1**	94.3
NVA	**93.3**	92.6	**95.6**	**95.8**	**94.1**	94.5
NA	92.8	92.4	95.6	95.4	93.9	**94.8**
VA	92.0	91.4	93.7	93.7	91.7	91.4

TAB. 7 – *Résultats du corpus C avec les différents descripteurs (précision).*

Annexe 2

Nous donnons l'espace de représentation d'HYBRED (tableau 8).

	Espace de représentation Avec HYBRED								
	NV			NVA			NA		
	3-cara	4-cara	5-cara	3-cara	4-cara	5-cara	3-cara	4-cara	5-cara
Corpus A	1288	1603	1822	1497	2027	2369	1393	1836	2092
Corpus B	1995	3294	4030	2339	4083	5188	2277	3788	4683
Corpus C	846	876	832	1000	1101	1083	906	929	901

TAB. 8 – *Tableau de comparaison de l'espace de recherche avec HYBRED.*

Annexe 3

Nous donnons les résultats obtenus avec l'approche HYBRED sur les corpus A (tableau 9) et C (tableau 10).

Algorithme K-PPV						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	75.6	78.4	72.8	71.2	71.2	77.6
3-caractères	92.0	**92.0**	94.0	95.2	94.0	90.4
4-caractères	**94.8**	86.0	**96.8**	**98.0**	**96.8**	**90.8**
5-caractères	94.4	85.2	95.2	96.4	94.0	90.4
Algorithme SVM						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	90.0	81.6	88.8	84.4	86.4	87.2
3-caractères	**97.6**	91.2	98.0	97.6	**98.4**	96.4
4-caractères	**97.6**	**95.2**	**98.4**	**98.4**	98.0	**98.0**
5-caractères	96.8	94.0	98.0	**98.4**	98.0	96.4
Algorithme NB						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	77.6	76.4	78.4	76.8	76.8	77.6
3-caractères	93.6	88.8	90.4	92.0	91.6	91.6
4-caractères	**94.8**	**91.2**	**93.6**	**94.8**	96.0	**92.4**
5-caractères	92.8	90.8	92.0	94.0	**96.4**	91.6

TAB. 9 – *Précision obtenue avec l'approche HYBRED pour le corpus A*

Algorithme K-PPV (TF.IDF)						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	87.5	67.1	84.3	90.6	84.3	79.6
3-caractères	96.8	**82.8**	96.8	98.4	96.8	**87.5**
4-caractères	**100**	75.0	**100**	**100**	**100**	78.1
5-caractères	**100**	81.2	**100**	**100**	**100**	81.2
Algorithme SVM (TF.IDF)						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	95.3	85.9	93.7	93.7	93.7	90.6
3-caractères	**100**	95.3	**100**	**100**	**100**	96.8
4-caractères	**100**	**96.8**	**100**	**100**	**100**	**100**
5-caractères	**100**	**96.8**	**100**	**100**	**100**	**100**
Algorithme NB (TF.IDF)						
Descripteurs	N	V	NV	NVA	NA	VA
2-caractères	87.5	67.1	**93.75**	**93.7**	**93.7**	**78.1**
3-caractères	76.5	70.3	84.3	79.6	84.3	76.5
4-caractères	**78.1**	70.3	79.6	73.4	79.6	71.8
5-caractères	76.5	**73.4**	71.8	75.0	71.8	65.6

TAB. 10 – *Précision obtenue avec l'approche HYBRED pour le corpus C*

Summary

The classification of digital documents is a complex task in a document analysis flow. The quantity of documents resulting from the OCR retro-conversion (optical character recognition) makes the classification task harder.

In the literature, different features are used to enhance the classification performance. In this paper, we evaluate various features of OCRed documents and non OCRed documents. Thanks to this evaluation, we propose HYBRED (**HYB**rid **RE**presentation of **D**ocuments) approach which combines different features in a single relevant representation. The experiments conducted on real data show the interest of this approach.