Université Sorbonne Nouvelle –Paris 3

Mémoire de Master 2 Traitement Automatique des Langues

Parcours Recherche & Développement

Classification par apprentissage automatique supervisé et hors ligne :

Application à la veille sanitaire

Encadrants
Mathieu Roche
Sulvain FALALA

Directeur
Jean-Luc Minel

Présenté par
Ibrahim BOMPOKO EBENGO

Stage effectué à
l'Unité Mixe de Recherche ASTRE
(INRA/Cirad)

Mars-septembre 2017

TABLE DES MATIERES

Liste des abréviations

ACP : Analyse en Composante Principale

AFNOR : Association Française de Normalisation

AGNES : AGlomerative NESted clustering

ALD : Analyse Linéaire Discriminante

ANSES : Agence Nationale de Sécurité Sanitaire

ASTRE : Animal, Santé, Territoires, Risques et Ecosystèmes

C5.0 : Classification supervisé 5.0. (Algorithme de type Arbre de décision)

CART : Classification And Regression Trees (Algorithme de type Arbre de décision)

CCP : Cirad Classifieur PadiWeb

CDC : Center for Disease Control

CDD : Contrat à Durée Déterminée

CHAID : CHi-squared Automatic Interaction Detector (Algorithme de type arbre de décision)

Cirad : Centre de coopération internationale en recherche agronomique pour le développement

Df : Document frequency

DGAL : Direction générale de l'alimentation

DIANA : DIvisiveANAlysis

ECDC : European Centre for Disease Prevention and Control

ESA : European Space Agency

FAO : Food and Agriculture Organization

FCO : Fièvre Catarrhale Ovine

GDT : Grand Dictionnaire Terminologique

INRA : Institut national de la recherche agronomique

Isomap : Isométriques map (Cartographie des Caractéristiques Isométriques)

J48 : Arbre de décision (Algorithme de Classification supervisé)

MARS : Multivariate adaptive regression splines (Régression multivariée par spline adaptative)

MVS : Machine à Vecteur de Support

OIE : Organisation mondiale de la santé animale

OMS : Organisation mondiale de la santé

OQLF : Office Québécois de la Langue Française

PMD : Positionnement Multi-Dimensionnel

PPA : Peste Porcine Africaine

ROC : Receiving Operator Characteristic (mesure de la performance d'un classifieur)

RSS : Really Simple Syndication (type de format pour la Syndication de contenu)

SBE : Système de Biosurveillance basé sur les événements

SVM : Support Vector Machines

Tf : Term frequency

TF.IDF : Term Frequency Inverse Document Frequency

TT4F : TreeTagger for Java

VC : Vapnik-Chervonenkis

VSI : volontariat de solidarité internationale

XML : Extensible Markup Language («langage de balisage extensible » en français)

Liste des tableaux

Liste des figures

INTRODUCTION GENERALE

La quantité disponible d'information sur le web, et les évolutions de la technologie, ont facilité l'acquisition et le recueil de nombreuses données notamment dans le domaine de l'intelligence épidémiologique.

Ces données permettent la constitution de corpus d'étude et peuvent servir de support conduisant au développement de système de surveillance, plus particulièrement le système de biosurveillance fondé sur les événements (le SBE) repérés à partir des articles de média web.

Ainsi plusieurs systèmes de surveillance épidémiologique ont été développés à cet effet (Systèmes développés depuis 2006, essentiellement en santé humaine (OMS, ECDC(Europe), Institute for Public Health (Canada), CDC (USA)).

Il existe des systèmes qui utilisent de combinaison de termes tels que le nom des maladies et les termes liés aux signes cliniques regroupés en syndromes, et d'expression de recherche, depuis de nombreuses années, à partir de sources non-officielles sur le web, tels que les flux RSS, les agrégateurs de nouvelles ou pages web.

Par ailleurs, les systèmes de surveillance de propagation des maladies infectieuses fondés sur les techniques d'intelligences artificielle, l'apprentissage automatique ou machine (Machine Learning, en anglais), sont de plus en plus envisagés.

Ils permettent, par exemple, d'améliorer les méthodes automatiques qui identifient des documents pertinents pour la tâche de veille épidémiologique. Ceci constitue l'objet principal de ce stage.

L'apprentissage machine ou automatique est un ensemble d'outils statistiques ou géométriques et d'algorithmes qui permettent d'automatiser la construction d'une fonction de prédiction (d'approximation) à partir d'un ensemble d'observations que l'on appelle ensemble d'apprentissage (Lemberger et al., 2016)

Ainsi lorsqu'il s'intègre au système de surveillance épidémiologique, le processus d'apprentissage utilise un ensemble d'observations (les descripteurs) reflétant l'événement sanitaire afin de construire un modèle de fonction de prédiction (d'en extraire les règles) de manière à permettre ultérieurement une prise de décision automatique, ou une prédiction (par exemple, identifier des documents propres à l'émergence d'une maladie).

Une prédiction correspond, alors, à l'évaluation de la fonction de prédiction sur les variables prédictives (les descripteurs) d'une observation donnée.

Chaque observation est décrite au moyen de deux types de variables : les variables prédictives (appelées aussi « attributs », « paramètres », dites encore « indépendantes »,

« de contrôle », « exogène » ou « observées ») et les variables cible (appelées aussi variable «à expliquer », « à prédire », « réponse », « dépendante », ou encore « endogène »).

Le critère le plus important parmi les types d'apprentissage est la distinction entre l'apprentissage supervisé (classement et techniques prédictives, nécessite que l'on dispose d'un ensemble d'exemples, caractérisés par des descripteurs dont on connait par avance les variables à prédire, l'étiquetage préalable) et l'apprentissage non supervisé (le clustering ou la classification, ne nécessite aucun étiquetage préalable de données d'apprentissage).

Parmi les techniques prédictives (de l'analyse ou fouille de données), nous avons le classement (ou la discrimination, la catégorisation, dite aussi « de classification » chez certains auteurs) et la régression (ou la prédiction) :

- Le classement est une opération qui permet de placer chaque individus de la population étudiée dans une classe, parmi plusieurs prédéfinies, en fonction des descripteurs (des caractéristiques indiquées comme variables explicatives). C'est un modèle d'apprentissage machine dont la cible est variable qualitative (catégorie ou classe).
- La régression, est un modèle d'apprentissage machine dont la cible est une variable quantitative (numérique).

Nous distinguons également l'apprentissage hors ligne (ou statique quand on connait en avance l'intégralité des données avant de procéder à l'apprentissage) et l'apprentissage incrémental (ou online, quand on se retrouve dans une situation où il existe un flot continu d'informations auxquels l'algorithme doit s'adapter, s'ajuster au fur et à mesure des données qui lui parviennent).

CONTEXTE

La veille en santé animale, et notamment la détection précoce d'émergences au niveau mondial d'agents pathogènes, est l'un des moyens permettant de prévenir l'introduction de dangers sanitaires dans un pays (en France, dans notre cas).

La principale plateforme d'épidémio-surveillance en France est la plateforme ESA, qui regroupe des instituts comme le Cirad, la DGAL et l'ANSES. Elle dispose d'une cellule chargée de la Veille Sanitaire Internationale (VSI).

La veille sanitaire pour cette plateforme est caractérisée par la recherche, l'acquisition, l'analyse et la communication d'informations épidémiologiques.

Les données épidémiologiques proviennent de sources officielles, qui correspondent à la déclaration des maladies par des organismes internationaux (OIE, FAO), et de sources non officielles comme les médias du web.

La veille basée sur des sources non officielles est manuellement effectuée par un analyste et est chronophage. Elle comprend la recherche manuelle sur le web, la lecture individuelle des différents articles de média, la sélection d'un contenu pertinent, ainsi que la préparation des rapports avec les dangers sanitaires repérés.

Pour optimiser cette veille manuelle, l'analyste utilise une technique qui s'appuie sur une collection de centaines de sujets et de mots-clés prioritaires qui lui permettent des requêtes automatiques sur le web (notamment sur Google news) et de préparer des synthèses quotidiennes résumant le titre de l'article acquis, la date, le lien, la source du contenu de la page web. Malgré ces optimisations, cette veille manuelle reste chronophage et elle repose sur la présence de l'analyste et son absence peut bloquer le processus de veille.

Afin de paliers ces difficultés, le Cirad, l'Anses et la DGAl développent depuis 2013 (thèse d'Elena Arsevska 2013-2017) un système de veille automatique du Web (baptisé « Padiweb ») qui effectue : (1) le recueil quotidien de dépêches épidémiologiques provenant de sources non officielles, incluant les médias électroniques, (2) l'extraction automatique d'informations (nom de maladie ou symptômes, lieu, date et espèce touchée) issues de ces dépêches et (3) une restitution synthétique et agrégée de l'information : cartes, séries spatiotemporelles. Actuellement, cinq maladies animales exotiques sont ainsi surveillées, mais d'autres pourraient l'être aisément, car l'outil est développé de façon générique.

Après l'étape de recueil des articles, une classification est nécessaire car seuls les articles traitant de l'apparition de nouveaux foyers intéressent les épidémiologistes.

Le but du stage est ainsi de développer, évaluer et intégrer un classifieur automatique dans la plate-forme Padiweb afin d'identifier les documents pertinents à traiter.

L'objectif de ce travail est donc de réaliser une méthode de classification fondée sur une liste de descripteurs les plus pertinents afin d'enrichir le système de filtrage de documents pertinents.

Ainsi, ce mémoire rend compte du processus de catégorisation automatique de textes par apprentissage supervisé et son intégration (en remplacement du filtre actuelle simplement fondé sur l'utilisation de mots-clés) dans la plate-forme logicielle Padiweb. Cette plate-forme est développée par le Cirad et l'INRA au sein de l'unité ASTRE et en collaboration avec l'unité TETIS.

La classification (le classement) automatique de document a pour but de rechercher un modèle mathématique de prédiction, une liaison fonctionnelle entre une collection de documents et un ensemble de classes (étiquettes ou catégorie).

Cette liaison fonctionnelle ou modèle de prédiction est estimée par un apprentissage automatique (traduction de machine Learning).

Pour mettre en place le processus de classification automatique, il est nécessaire de disposer d'un corpus d'apprentissage à partir duquel nous estimons les paramètres du modèle de fonction ou prédiction de classes le plus performant possible, c'est-à-dire le modèle qui produit le moins d'erreur en prédiction des classes.

L'intérêt d'une telle démarche permet d'organiser les connaissances, les articles de médias dans notre cas, de les regrouper de manière efficace en classe, de façon à limiter les bruits dans une collection de documents (extraits du web dans notre cas), afin de pouvoir effectuer, par la suite, des tâches telles que l'extraction ou recherche d'information efficace.

METHODOLOGIE

L'approche que nous avons mise en œuvre se décline en quatre étapes : la représentation textuelle, la sélection de variable (de descripteurs), le choix de modèle du classifieur et la classification de classes des documents.

Ces quatre étapes de notre approche suivent deux grandes phases : la construction d'un modèle de fonction de prédiction (la phase d'apprentissage), et le classement (la phase de prédiction de classe).

La première, la phase d'apprentissage :

- Nous disposons d'un corpus étiqueté (pour chaque texte ou article, nous connaissons, en avance sa classe) ;
- A partir de ce corpus, nous extrayons les descripteurs (dites aussi « variable prédictive », ou « attributs ») (le sous-ensemble de variables) les plus pertinents ;
- Nous disposons alors d'une matrice « attributs x articles » poids (pour chaque article nous connaissons le poids (ou valeur) de ses attributs et sa classe (son étiquette) ;
- Nous appliquons ensuite l'algorithme d'apprentissage sur cette matrice afin de construire et d'obtenir un modèle de fonction de prédiction.

La seconde, la phase de prédiction de classe (le classement) :

- Recherche les pondérations d'un nouvel article à classer ;
- A partir de ces pondérations, le modèle de fonction de prédiction va prédire l'étiquette ou la classe de ce nouvel article.

OBJECTIFS

Nous proposons dans ce travail, une approche (modèle de classification ou de prédiction de classe) comme outil d'amélioration de la qualité du système de veille (Padiweb dans notre cas).

A partir d'un ensemble de données et d'une terminologie existants, nous construisons un modèle (un classifieur), en tenant compte des spécificités de l'environnement du logiciel.

Notre modèle performe bien et se caractérise par les propriétés suivantes : une bonne capacité prédictive, une facilité d'interprétation et une capacité suffisante de généralisation.

La première propriété permet l'adéquation du modèle avec l'environnement considéré (le web dans notre cas, notamment dans le cadre de la veille sanitaire). La seconde propriété offre la possibilité de maintenir le logiciel. Enfin, la troisième permet au modèle d'être robuste (en d'autres termes, de tenir compte de l'évolution future du contexte logiciel dans lequel il est utilisé).

PROBLEMATIQUES

La problématique dont traite ce travail consiste à la classification par apprentissage supervisé et son intégration dans le système de biosurveillance.

Les difficultés particulières auxquelles fait face l'apprentissage automatique supervisé (le classement ou la catégorisation de texte) sur des angles différents sont :

- D'une part (liées au données), les grandes dimensions (qui influent sur la performance d'algorithme), l'imprécision des fréquences, les déséquilibrages (de données de classes dans le corpus), l'ambiguïté, la synonymie, la subjectivité (catégorie ou classe attribuée en fonction du contenu sémantique de texte, et dépendant du jugement d'expert) (Radwan, 2003);

- Et d'autre part liées (liées au modèles) la robustesse, la concision, les résultats explicites, la diversité des types de données manipulées (toutes les technique ne sont pas apte à manipuler les données manquantes, les données qualitative, etc.), la rapidité de calcul du modèle et les possibilités de paramétrage(Stéphane, 2012a).

Ainsi, dans ce mémoire, nous nous efforçons de répondre à ces différentes questions scientifiques en exploitant les théories, méthodes et outils existant dans la littérature, afin de développer un classifieur original tenant compte de ces différentes propriétés.

PRESENTATION D'ASTRE

J'ai effectué mon stage au sein de l'unité ASTRE (Animal, Santé, Territoires, Risques et Ecosystèmes), unité mixte de recherche INRA/Cirad créée pour la recherche intégrée en Santé animale.

Missions

Les objectifs de l'unité ASTRE, sont entre autres :

- D'améliorer la santé animale, la santé et la sécurité alimentaire au Sud notamment dans le cadre des changements globaux et des transitions des socio-écosystèmes ;
- De développer une approche intégrative de la santé : inter-sectorialité et interdisciplinarité autour de l'animal, de l'écosystème et du territoire, dans le cadre des approches On Health et EcoHeath.

Les axes de recherche

L'unité Astre, travaille sur deux axes stratégiques de recherche :

- La catégorisation des pathosystèmes et des épisystèmes, au niveau de l'organisme, de la population et du socioécosystème ;
- La recherche en gestion de la santé : la surveillance (innovation, évaluation et réseaux) et le contrôle (outils et stratégies, action collective) des maladies.

Des ressources multi-disciplinaires en synergie

- Une centaine d'agents permanents (chercheurs, ingénieurs, techniciens, administratifs) et plus d'une trentaine de non permanents (doctorants, volontaires de service civique, CDD…) ;
- Des champs disciplinaires complémentaires : pathologie, immunologie, génomique, virologie, bactériologie, entomologie, écologie, épidémiologie, modélisation, anthropologie, géographie, sociologie…

Implantation mondiale

France : Montpellier ; Caraïbe : Guadeloupe ; Océan Indien : Réunion, Mayotte, Madagascar ; Afrique : Maroc, Sénégal, Ethiopie, Zimbabwe, Mozambique, Afrique du Sud ; Asie du Sud Est : Thaïlande, Cambodge, Laos, Vietnam.

ASTRE est en partenariat au Sud et au Nord avec des organismes de recherche, Universités, Services vétérinaire ou santé humaine, organisations nationale et régionales, Agences.

ETAT DE L'ART DE CLASSIFICATION AUTOMATIQUE DE DOCUMENT

Ce chapitre se propose de présenter brièvement les familles de méthodes de classification et de réduction de dimension de l'espace de représentation pour la classification de textes.

1. La classification automatique de documents par apprentissage

Nous distinguons principalement deux méthodes d'apprentissage : l'apprentissage numérique et l'apprentissage symbolique.

Notre travail s'inscrivant dans le cadre de théorie d'apprentissage numérique, dans cette partie du manuscrit, nous allons passer en revue, les méthodes et techniques de classification automatique.

Dans la littérature, il existe des méthodes ou techniques de classification automatique non supervisées (le clustering, ou le regroupement), et des méthodes ou techniques de classification supervisées (le classement et de prédiction).

1.1. La classification automatique non supervisée (méthode descriptive)

La classification automatique non supervisée (dites « regroupement », ou « clustering » en anglais, on lui donne souvent le nom de « segmentation ou typologie » en marketing, « nosologie » en médecine, « taxinomie ou taxonomie » en biologie) est une méthode issue de la statistique et notamment de l'analyse de données.

Elle permet de regrouper les observations (les individus, les variables, ou les objets) présentant des traits communs, sans superviseur comme son nom l'indique (l'appartenance des objets ou individus aux familles ou classes n'est pas connue à priori). En fonction des caractéristiques similaires d'individus de la population ou de l'échantillon étudié, le programme (le classifieur) découvre par lui-même les structures (les relations non spécifiées) plus ou moins cachées de données. Cela vient de ce qu'il n'y pas de variable à prédire.

En effet, la classification est une méthode descriptive et non prédictive.

Elle permet simplement l'extraction des similarités, à partir d'une population ou d'un échantillon analysé, des variables ou groupe d'individus présentant des traits communs (les données similaires), le nombre, la structure ou la définition des familles (classes, partitions, strates, ou « cluster » en anglais) n'étant pas toujours connus en avance mais découverts durant le processus de regroupement, contrairement aux classes de classement.

Au sens où nous l'entendons dans cette partie du manuscrit, la classification fait référence à l'apprentissage non supervisé.

Sachons qu'avec les statisticiens français, la classification (le clustering), au sens large, peut faire référence aux approches supervisées où on connaît à priori les structures des classes à partir données d'apprentissage(Lutz and Biernat, 2015).

1.1.1. Les principales approches ou méthodes

Selon les résultats produits, deux grandes approches de regroupement automatiques des objets ou individus sont identifiées :

- La **classification hiérarchique** consiste à construire une suite de partitions emboîtées les unes dans les autres. Les relations entre partitions de différents niveaux peuvent ainsi être représentées sous forme arborescente appelée dendrogramme.

  Deux familles de méthodes sont distinguées (**ascendants**, méthodes les plus populaires, qui construisent par agglomérations successives des objets deux à deux (AGNES pour AGlomerative NESted clustering) ; et **descendants** qui réalisent des dichotomies progressives de l'ensemble des objets (DIANA pour DIvisiveANAlysis)).

- La **classification non hiérarchique** (dites aussi « partitionnement ») consiste à repartir les individus en différentes partitions, pour un nombre de partitions fixé, chaque individu n'appartient qu'à une seule partition (ou groupe).
  Dans cette famille des méthodes, on retrouve **l'approche partitive** (dont le résultat est une partition en un nombre de groupe fixé, donné, ou découvert par l'algorithme. Deux familles sont identifiées, **les approches basées sur les prototypes** (les K-Moyenne) et **les approches basées sur le voisinage** (la carte de Kohonen)) et **floue et probabiliste** (permet d'attribuer des valeurs de probabilités d'appartenance des individus à chaque groupe, ce qui a pour effet d'obtenir une partition floue des individus (K-Moyenne floue)).

Selon la spécificité de chaque méthode de classification, trois grandes approches sont identifiées :

- **L'approche discriminative** vise à déterminer géométriquement des frontières de décision de séparation ou regroupement des individus dans un espace donné ;
- **L'approche probabiliste** dont le degré d'appartenance ou l'affectation des individus dans des classes est fondée sur un modèle probabiliste, comme son nom l'indique, l'individu est affecté au groupe le plus probable.
- **L'approche heuristique ou algorithmique** dont les partitions ou groupes sont obtenus par pure recherche heuristique.

1.1.2. Les étapes de méthodes de classification hiérarchique

Afin de regrouper les objets, la méthode de classification hiérarchique passe par les étapes suivantes :

A partir des observations à regrouper ;

1. On construit un tableau (une matrice) des distances entre les observations et on cherche les deux plus proches que l'on agrège en un nouvel élément. On obtient un premier regroupement à n classes (où n est l'effectif des observations moins un) ;
2. On construit une nouvelle matrice des distances qui résulte de l'agrégation précédente, en calculant les distances entre le nouvel élément et les observations restantes, les autres distances restent inchangées. On se retrouve dans les mêmes conditions à l'étape où assertion 1, mais avec effectif des observations moins un, et en ayant choisi un critère d'agrégation. On cherche de nouveau les deux observations les plus proches, que l'on agrège. On obtient un deuxième regroupement avec effectif des observations moins deux, qui englobe le premier regroupement ;
3. On calcule les nouvelles distances, et l'on réitère le processus jusqu'à n'avoir plus qu'un seul élément regroupant toutes les observations : ceci représente le regroupement final.

Le principe des méthodes ascendantes et descendantes peut être résumé de la manière suivante :

Algorithme approche ascendante

- Partition initiale : chaque objet forme une classe ;
- Pour i allant de 1 à l'EffectifObjets – 1 :
    - Choisir et fusionner deux classes parmi les l'EffectifObjets – i+1

Algorithme approche descendante

- Partition initiale : tous les objets dans une même classe ;
- Pour i allant de 1 à l'EffectifObjets – 1 :
    - Choisir une classe à scinder parmi les i ;
    - Scinder la classe

1.1.3. Mesure d'éloignement

Les algorithmes de classification non supervisés sont tous fondés sur des mesures permettant de quantifier l'éloignement ou la proximité entre les objets ou individus.

Parmi les diverses mesures existantes, nous avons :

- La distance ;
- La dissimilarité (ou dissemblance);
- Et la similarité (ou ressemblance).

La similarité et dissimilarité permettent de mesurer le lien entre les individus d'un même ensemble ou entre les variables.

Domaine d'applications

Les algorithmes de classifications ont été appliqués dans des domaines ou applications suivants :

- Analyse d'ADN, dans le domaine de la bio-informatique ;
- Découverte de classes de patients présentant des caractéristiques physiologiques commune, dans le domaine de la médecine ;
- Segmentation ou typologie des clients, en marketing ;
- Etc.

1.2. La classification supervisé /méthode prédictive

La classification (le classement) permet de prédire, en fonction des descripteurs d'entrées (attributs prédictifs), une valeur d'attribut « à expliquer ou à prédire » qui prend ses valeurs dans un ensemble fini d'étiquettes (dites aussi « classes »), à partir d'un ensemble de données étiquetées (couples d'exemples d'observations et les réponses associées).

Dans le cadre de ce mémoire, notre approche s'inscrit dans le cadre de techniques de classement et prédiction.

Les principales approches ou méthodes

Plusieurs méthodes, modèles prédictifs ou algorithmes de classement existent : les arbres de décisions (CART, C5.0, CHAID, etc.), ou forêt aléatoires, l'algorithme MARS, les méthodes bayésiennes (dit « Naïf Bayes »), le SVM, les réseaux de neurones et les algorithmes génétiques, etc.

Dans ce chapitre, nous nous concentrons sur les trois algorithmes les plus utilisés dans la littérature, à savoir :

- Les Arbres de décision ;
- Naïf Bayes
- et SVM.

1.2.1. Les Arbres de décisions

Les arbres de décision sont parmi les techniques de classement les plus populaires et intuitives.

1.2.1.1. Descriptions

Les arbres de décision sont des ensembles de règles de classification fondant leur décision sur des critères ou test associés aux attributs ou variables explicatives, organisés sous forme d'arborescence.

Elles sont composées de nœuds de décision (nœuds internes étiquetés respectivement par un test pouvant être appliqué à chaque attribut d'un individu ou d'une population), qui permettent de tester les variables prédictives, elles supportent bien les données hétérogènes, manquantes et les effets non linéaires. En effet, elles sont extrêmement flexibles.

Elles contiennent de branches (arcs issus d'un nœud interne, réponse possible au test du nœud) qui représentent chacune une valeur de variable explicative testé, elles peuvent être constituées de nœuds terminaux (de feuilles) représentant les classes résultantes.

L'idée principale de ces techniques consiste, au sens informatique, à classer un individu au moyen d'une succession de questions (ou critère de classement) concernant la valeur des variables explicatives d'observation (Lemberger et al., 2016).

Ainsi, pour la construction d'un arbre de décision destiné au classement des observations, on opère un choix de variable qui sépare le mieux les individus de la population de chaque classe, le critère (de Gini, Twoing, du chi-deux, l'entropie ou information) de choix de la variable et de la condition de segmentation ou séparation sur cette variable dépend de chaque type d'arbre (CART, CHAID, C4.5 et C5.0).Le critère de Gini et de Twoing sont utilisé dans l'arbre CART ;Le critère du chi-deux est utilisé dans CHAID ; l'entropie ou l'information est utilisée dans C4.5 et C5.0

1.2.1.2. Type d'algorithmes d'arbres

Les principaux types d'arbres, présentés dans la littérature, sont :

- L'arbre CART (Classification And Regression Tree) inventé en 1984 par les statisticiens d'universités de Berkeley et de Standford, est adapté à l'étude de tout type de variable, est l'un des arbres les plus efficaces et les plus répandus (Il s'appuie sur le critère ou indice de Gini, pour trouver la meilleure segmentation ou séparation de chaque nœud) ;
- L'arbre CHAID, est une des conceptions plus anciennes (il utilise le test de chi-deux pour définir la variable la plus significative de chaque nœud).

- L'arbre C4.5 et C5.0, sont des perfectionnements de précédents arbres, par des chercheurs (ils fonctionnent en cherchant à maximiser le gain d'information réalisé en affectant chaque individu à une branche de l'arbre).

1.2.1.3. Avantages et inconvénients

Tout arbre de décision définit un classifieur qui se traduit immédiatement en termes de règles de décision fondées sur des méthodes purement algorithmiques (qui ne s'appuient sur aucun modèle probabiliste).

L'avantage des arbres de décisions est qu'ils fournissent des règles explicites de classement, les décisions sont aisément interprétables par l'utilisateur, la classification est très rapide.

Néanmoins, le critère de séparation ou de segmentation affecté au premier nœud possède une très grande influence sur le modèle de fonction de prédiction, si bien qu'un ensemble de données peu représentatif pourra mener à des conditions totalement erronées ; il existe de risque de surapprentissage (liée à l'existence de la variable à prédire, sous une forme déguisée, parmi les variables prédictives) si l'arbre n'est pas correctement élaguée (Lemberger et al., 2016), etc.

1.2.2. Les méthodes bayésiennes

1.2.2.1. Le classifieur naïf bayes

La méthode de classement Naïf Bayes est une méthode initialement développée pour des taches de classification qui repose sur le théorème de Bayes avec une forte indépendance (dite naïve, c'est-à-dire elle suppose que l'existence d'une variables de contrôles pour une classe donnée est indépendante conditionnellement à l'existence d'autres variable de contrôles) des hypothèses, même si les variables sont liées dans la réalité.

En dépit d'hypothèse indépendante forte, la méthode de naïf Bayes constitue un outil très efficace de classement sur de nombreuses applications réelles. Elle est particulièrement efficace lorsque la dimension (le nombre de variables en entrée) de l'espace des variables de contrôle est élevée.

Le classifieur naïf bayes s'étend parfois à des situations où les relations de dépendances conditionnelles sont plus complexes, les modèles adjoints s'appellent réseaux bayésiens.

1.2.2.2. Les réseaux bayésiens

Le réseau bayésien est un graphe causal auquel on a associé une représentation probabiliste sous-jacente. Ce réseau permet donc de quantifier le raisonnement sur les causalités que l'on peut faire à l'intérieur du graphe, l'ensemble de probabilités « conditionnelles » associées est aussi défini. Il se compose donc de deux éléments : la représentation de dépendance d'une part, et la quantification ou la mesure de ces dépendances.

Ces réseaux doivent leur nom aux travaux de Thomas Bayes au XVIIIe siècle sur la théorie des probabilités, sont le résultat de recherches effectuées dans les 1980, dues à J. Pear et une équipe de recherche danoise (Naïm et al., 2011).

Dans le classifieur naïf bayes, la variable à contrôler est parente de chacune de variable des contrôles, mais une variable à contrôler n'est jamais parente d'une autre. Cela se traduit dans le réseau bayésien par un graphe dont les variables de contrôle sont toutes au même niveau et jamais reliées entre elles par un lien.

Avantages et inconvénients

La méthode de classification bayésienne est simple à utiliser et à interpréter, elle se programme assez aisément, sa simplicité lui rassure de bonnes performances.

Le classifieur naïf bayes peut présenter un intérêt, en fournissant une base de référence pour les performances des autres modèles. En dépit de naïveté de ses hypothèses, il est parfois étonnamment performant (Stéphane, 2012b).

L'intérêt des réseaux bayésiens est qu'ils permettent de représenter graphiquement les connaissances (dites aussi « boîtes blanches », par opposition à la « boîte noire » de réseaux de neurones), qu'ils sont faciles à utiliser et à modifier, et qu'ils permettent de faire des inférences.

Néanmoins, les prédictions de probabilités pour les différentes classes sont erronées lorsque l'hypothèse d'indépendance conditionnelle est invalide (Lemberger et al., 2016).

1.2.3. Le Support Vecteur Machine (SVM)

Le SVM est un des algorithmes les plus performants en classification de textes. Il a été proposé par Vapnik dans son livre « The nature of statistical leaning theory ». Il repose sur deux notions clés : la notion de marge maximale et la fonction noyaux.

L'idée principale de l'algorithme est donc de trouver une frontière de décision (l'hyperplan) qui sépare au mieux les données dans l'espace de représentation et dont la séparation est aussi grande que possible. L'hyperplan calculé permet ainsi de séparer l'espace de représentation en deux régions ou zones.

Pour classer les nouveaux articles ou documents, on calcule dans quelle région de l'espace ils se situent et on leur attribue ou associe la classe correspondante.

La plupart des méthodes de classification utilise la représentation textuelle qui rencontre les problèmes de la dimensionnalité d'espace de représentation de textes. Ce point sera abordé dans la section suivante.

La réduction de la dimensionnalité pour le classement de textes

Dans la littérature, deux grandes familles de méthodes ont été identifiées, à savoir la réduction fondée sur l'extraction de descripteurs les plus adaptés et la réduction fondée sur une sélection de descripteurs.

La réduction fondée sur l'extraction de caractéristique est composée de deux méthodes : linéaire (ACP pour Analyse en Composante Principale, ALD pour Analyse Linéaire Discriminante, et PMD pour Positionnement Multi-Dimensionnel) et non linéaire (Isomap, et plongement linéaire).

Comme notre algorithme est fondé sur la sélection de sous-ensembles, dans cette section, nous présentons principalement l'approche de sélection de descripteurs.

Sélection de descripteurs

La sélection de descripteurs permet de trouver un sous-ensemble « pertinent » d'attributs parmi ceux de l'ensemble original de termes (dit aussi ensemble de départ) pour la prédiction de la variable à expliquer ou à prédire. C'est un aspect essentiel de l'apprentissage supervisé pour différentes raisons : la simplicité, le déploiement facile, la généralisation ou la robustesse du modèle. La figure 1, illustre bien ces opérations.

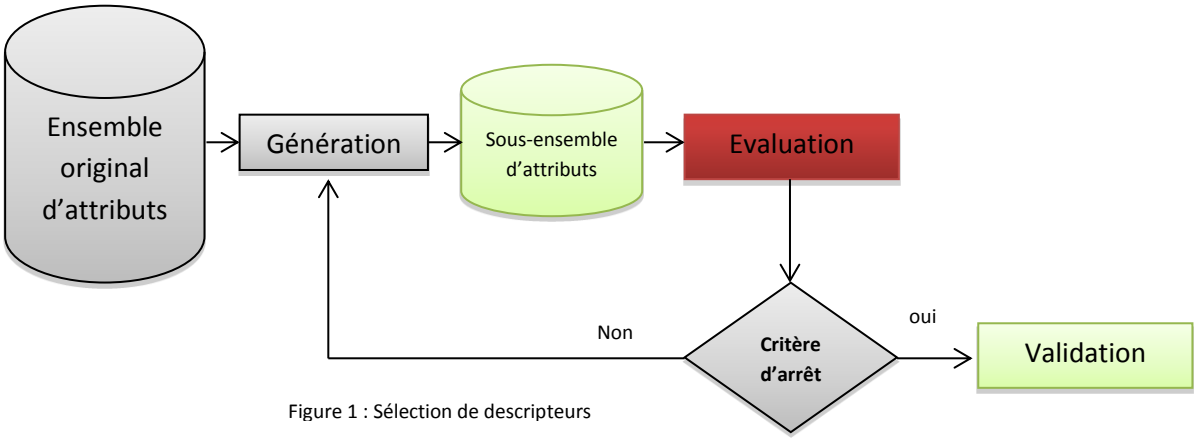

Figure 1 : Sélection de descripteurs

Catégorisation de méthode de sélection de descripteurs

Pour catégoriser les sélections d'attributs, trois axes suivants ont été identifiés : les stratégies ou méthodes de recherche (séquentielle, aléatoire, et complète), les critères d'évaluation (Filter, Wrapper, Embeded) et les algorithmes de classification (Liu et Yu en 2005).

Type de stratégie (ou méthodes) de sélection de sous-ensembles

Trois grands types de stratégies ont été identifiées dans la littérature : (1) La première méthode, consiste à définir en avance la taille du sous-ensemble, ensuite l'algorithme de sélection va trouver le meilleur sous-ensemble en fonction de cette taille (prédéfinie); (2) La deuxième méthode, vise à sélectionner le plus petit sous-ensemble dont la performance est supérieure ou égale à un seuil défini en avance ; (3) Enfin la troisième méthode, est un compromis entre la réduction de la taille et la performance de classifieur.

Critères d'évaluation d'attributs

Les trois familles de méthodes de filtrage mises en avant dans la littérature sont : Filter (filtre en français), Wrapper (dite aussi« enveloppe » en français) et Embeded (intégré, en français).

Les méthodes filter (de filtrages) permettent d'enlever, à l'étape de prétraitement, les variables non pertinentes avant la phase d'apprentissage. En effet, elles permettent de sélectionner les variables pertinentes indépendamment d'algorithme d'apprentissage mise en œuvre par la suite, grâce à des mesures telles que (Document Frequency abrégé DF, Entropie, Information mutuelle, etc.).

Pour les méthodes Embeded (Intégré), les techniques de sélections d'attributs pertinents font partie de l'apprentissage. Les algorithmes d'induction d'arbres de décision les illustrent parfaitement bien.

Les méthodes Wrapper (enveloppe) utilisent l'algorithme d'apprentissage pour choisir le sous-ensemble d'attributs (parmi d'autres présentés à l'algorithme d'apprentissage) réellement discriminants, ou qui optimise le mieux, en employant explicitement le critère de performance (l'aire sous la courbe ROC, ou le plus souvent le taux d'erreur).

Critères de pertinence des attributs

L'algorithme d'apprentissage est fortement lié aux attributs, leurs présences ou absences peuvent fortement impacté les méthodes ou processus de classification.

Un attribut est jugé très pertinent si son absence entraîne une baisse significative de la performance du classifieur ; il est dit peu pertinent, s'il n'est pas d'abord jugé « très pertinent », et si son absence n'influence pas significativement la performance ou robustesse d'algorithme de classification ; enfin, il est jugé non-pertinent, lorsque son absence n'influence pas du tout la performance (en d'autres termes il est ni « peu pertinent » ni « très pertinent »).

CORPUS ET RESSOURCES

Ce chapitre présente le corpus de travail et les ressources linguistiques utilisés conjointement dans le cadre de notre étude, pour la construction d'un modèle pour la prédiction des classes.

Corpus

Cette partie du manuscrit consiste à la présentation les caractéristiques essentielles de notre corpus d'étude. Nous avons préalablement évoqué comme base conceptuelles, les définitions de corpus selon les acceptations des auteurs, les critères définitoires, et le type corpus.

DEFINITIONS

Le corpus est défini dans un cadre *large* (en attribuant le statut corpus à une simple collection de documents, si l'on se place comme le suggèrent Kilgarriff et Grefenshted) ou *précis* (en attribuant le statut corpus à une collection de documents (textes, images, vidéos, etc.) regroupés dans une optique précise).

En effet, en linguistique, les définitions de corpus sont souvent plus circonscrites. Comme l'explique Douglas Biber : « un corpus n'est pas seulement une collection », « même si toute archive de texte peut être en théorie un corpus, ajoute Mc Energy et Wilson ».

Ainsi, François Rastier, suggère la définition du corpus comme un regroupement structuré de textes intégraux, documentés, éventuellement enrichis par des étiquetages, et ressemblés : (i) de manière théorique réflexive en tenant compte des discours et des genres, et (ii) de manière pratique en vue d'une panoplie ou gamme d'applications (Rastier, 2002).

A côté de cette définition, John Sinclair, l'un des pères de la linguistique de corpus anglo-saxonne, en donne une définition particulièrement restrictive : « A corpus is a collection of naturally-occurring language text, chosen to characterize a state or variety of a language (Sinclair, 1991) . A collection of pieces of language that are selected and order according to explicit linguistic criteria in order to be used as sample of the language (Eagle, 1996) ».

Pour qu'un ensemble de textes mérite donc l'appellation corpus, il doit respecter aspects suivants (Bowker and Pearson, 2002) : un corpus est un assemblage important de textes authentiques compilés sous format électronique selon des critères précis.

LES CRITERES DEFINITOIRES

Les critères qui différencient un assemblage de données du corpus constitué sont, selon NajibArbach, les suivants :

- Le critère de la collection des données (notion qui sous-entend ou implique un protocole de collecte indiquant des objectifs linguistiques);
- Le critère du numérique (afin que l'archivage et l'analyse soient effectués par la machine informatique) ;
- Le critère de la représentativité des données (se résume par le potentiel de l'échantillon à représenter des vérités) ;
- Le critère de l'annotation des données (l'enrichissement ou l'ajout de données qui ne sont pas explicitement présents lors de l'assemblage ou compilation de données) ;
- Le critère de la documentation du corpus (assimilée à la description des données).

On retrouve également la notion de critère dans la définition selon Fisher : « Un corpus est donc un ensemble de textes (corpus textuel), un ensemble de mots (corpus lexical), ou un ensemble de phrases (corpus phrastique). Les textes sélectionnés sont selon des critères prédéfinis pour représenter, tant que possible, une langue, une variété de langue, un genre, un domaine de discours, un auteur, ou un sujet (…) »

Dans ce mémoire, nous avons travaillé sur une collection d'articles de media web (en format json) enrichis manuellement par des étiquettes.

SOURCES (ET TEMPS COUVERT PAR LES TEXTES) DU CORPUS

Les documents de notre corpus de travail ont été collectés manuellement (en août et septembre 2014) à partir de moteur de recherche de Google et de la base de données de la littérature biomédicale PubMed (est la plus grande base de données électronique en ce qui concerne la littérature biomédicale).

CONSTITUTION DE CORPUS

Le corpus étant composé uniquement d'articles référencés par Google, deux pratiques s'offraient ou s'opposent théoriquement, le «Web as Corpus : Web comme Corpus ou ressource linguistique» et le « Web for Corpus : le butinage du Web pour la constitution de Corpus», dans la littérature.

La première approche s'appuie sur l'utilisation de moteur de recherche général. Cette approche se réalise à moindre coût puisque l'essentiel du traitement est réalisé par le moteur de recherche commercial interrogé.

L'inconvénient avec cette pratique reste la limitation de nombre de pages à télécharger par jour ou une période donnée via l'api, elle souffre également de la limitation de pages référencées, ou indexées par le moteur de recherche tel que Google. Ces arguments amènent Adam Kilgarriff à la conclusion « Google ontology is bad science » (Adam, 2006).

Dans la littérature, cette approche reste utilisable quand on connaît les limites et qu'on est capable de tenir compte des biais qu'elle peut entrainer.

La deuxième approche (utilisée pour la constitution de notre corpus de travail) repose sur l'extraction des documents web particuliers répondant à des critères précis (dans notre cas nous avons utilisé des mots clés proposés par des experts du domaine).

Cette approche passe généralement par trois étapes : (i) la recherche sur le Web et l'extraction de documents (les articles) potentiellement pertinents ; (ii) la sélection des documents visités en fonction de leur pertinence par rapport à des critères prédéfinis ; et (iii) le filtrage des documents (articles) et leur stockage sous une forme utile pour la tâche envisagée (pour notre cas, la veille sanitaire). Un module de la plateforme PadiWeb a été réalisé à cette fin. Ainsi, pour l'acquisition des documents, on effectue des requêtes sur Google et PubMed en anglais, en utilisant la combinaison : « nom de la maladie » et « outbreak ».

TYPE DE CORPUS

En lien avec les définitions proposées, plusieurs genres de corpus ont été identifié (Bowker and Pearson, 2002), à savoir :

- **Corpus de référence** (considéré comme représentatif, permet la comparaison entre lui-même et un autre corpus afin de mesurer les écarts. Il fonctionne ainsi comme un corpus témoin. C'est un mélange de plusieurs documents ou textes de natures différentes, il permet ainsi de faire des observations d'ordre général) **et corpus spécialisé** (il est axé sur l'aspect particulier du vocabulaire d'un domaine, sur un certain type de textes, etc.) **;**
- **Corpus écrit** (contient des textes destinés à être lus) **et corpus oral** (constitué de transcriptions de discours oraux)**;**
- **Corpus monolingue** (rassemble des textes dans une langue) **et corpus multilingue** (rassemble des textes dans au moins deux langues, contrairement à un corpus monolingue). Les corpus multilingues peuvent être divisés en **Corpus parallèles** (constitué de textes en langue source accompagnés de leurs traductions en langue cible) et en **Corpus comparable** (ne contient pas des traductions mais de textes écrits en langues en langue sources qui ont tous le même sujet, la même fonction de communication, qui sont de même nature)**;**
- **Corpus synchronique** (est une photo de l'usage d'une langue ou d'une observation pendant un temps limité) **et corpus diachronique** (sert à mesurer l'évolution d'une langue ou d'une observation sur une longue période)**;**
- **Corpus ouvert** (est constamment étendu, il n'est pas figé, à l'opposé de corpus clos) **et corpus clos** (appelé aussi **corpus fermé ou statique**, est un corpus immuable. Une fois qu'il est constitué, aucun texte n'est ajouté) **;**
- **Corpus « apprenants »** (contient des textes écrits par les apprenants d'une langue étrangère)**.**

Le corpus mobilisé lors de nos travaux est une collection d'articles de media web (en format json) enrichis manuellement par des étiquettes.

Nous avons travaillé sur un corpus monolingue composé des textes anglais, un corpus spécialisé contenant de texte particulièrement envisagé pour une étude de la veille épidémiologique.

Il s'agit d'un corpus clos (aucun article n'a été ajouté jusqu'à la fin de notre étude) pour l'entrainement de notre algorithme d'apprentissage de modèle.

Il s'agit en outre d'un corpus écrit (les textes ont été produits dans le but d'être lus) et synchrone (dont les textes ou les articles ont été filtré dans un espace d'un mois en août et septembre 2014).

LA REPRESENTATIVITE DE DONNEES DU CORPUS

La question de représentativité est centrale à nos yeux, elle permet de lutter contre les généralisations abusives de notre modèle ou d'algorithme de classification. Dans la littérature, ces généralisations correspondent à deux types (Biber, 1993): la déformation et l'incertitude.

On parle de la déformation quand on utilise un échantillon dont les caractéristiques ne correspondent pas à celles de la population que cet échantillon est censé représenter ; et de l'incertitude lorsque le corpus est trop petit.

Nous nous retrouvant dans cette configuration avec le corpus de travail. On trouve par exemple moins de deux cents à trois cents occurrences d'article web de media par classe. Aussi les classes ne sont pas équilibrées.

EQUILIBRAGE DU CORPUS

Une autre question liée à la représentativité dans la littérature est celle de la notion de corpus équilibré. Un corpus représentatif est en effet un corpus équilibré.

Un corpus est dit équilibré lorsque la taille de ses sous-catégorie (genres, classes dans notre cas) est proportionnelle à leurs fréquences d'occurrence dans les observations. Ainsi, le nombre d'articles par classe devrait l'être de manière proportionnelle mais pas forcément en quantité égale par rapport à la population.

Dans notre cas, nous avons travaillé avec un écart entre les effectifs de chacune classes de notre corpus.

Cette question d'équilibre des strates du corpus, dans la littérature, est un problème épineux, liée à celle du point de vue de ses compilateurs ou auteurs.

LA TAILLE DU CORPUS

La question de la taille est liée à la question de la représentativité : plus le corpus est gros, plus il serait représentatif de la population qu'il vise à étudier. Le corpus doit évidemment atteindre une taille critique (généralement supérieure à des centaines d'individus dans chacun de classes à prédire) pour construire un modèle suffisamment robuste, ainsi pour permettre des traitements statistiques fiables.

Dans la littérature (Stéphane, 2012a), il est recommandé de disposer d'au moins 300 à 500 individus dans chacune des classes à prédire (en deçà de ce seuil, le modèle appris se généralisera mal faute d'un apprentissage parfaitement représentatif : le test doit se faire par validation croisée).

Cette propriété appelée consistance, en théorie d'apprentissage automatique, telle que vérifiée par Vapnik, n'est vraie que pour le processus correspondant aux familles de modèles dont la VC-dimension (pour Vapnik-Chervonenkis dimension) est finie. En effet, la VC-dimention d'un modèle est égale au nombre des paramètres dans quelque cas simple tel que les modèles linéaires. Les Support Vector Machine (SVM) sont l'un des premiers types de modèles dont il fut possible de calculer la dimension VC.

Mais dans la littérature, la taille d'un corpus ne garantit pas sa représentativité, selon certains auteurs McEnery et Wilson. Pour les traits fréquents, un petit corpus serait représentatif, alors que pour d'autres traits, la taille de l'échantillon doit être plus grande (Biber 2000) pour construire un modèle qui va permettre de prédire une situation.

Les chercheurs n'ont pas abouti, à l'heure actuelle, à un consensus en ce qui concerne la taille du corpus dans sa totalité. Ainsi la taille de notre corpus de travail est 23,0 Mo (24 159 111 octets), soit 25,2 Mo (26 484 736 octets) sur disque.

SPECIALISATION DE TEXTES DU CORPUS

Le corpus étudié dans ce mémoire est constitué d'articles de média qui ont été validés et catégorisés selon leur contenu comme :

- Pertinent (nouveaux cas) pour les articles de média décrivant l'apparition des foyers pour chacune des maladies modèles. Tous les documents pertinents traitent de sujets en rapport avec i) les foyers de la PPA en Europe de l'Est ou Afrique sub-saharienne, ii) les foyers de la FA en Afrique du Nord ou iii) les foyers de la FCO dans les Balkans, sur une période donnée entre 2011 et 2014 ;
- Non pertinent (bilan) pour les articles décrivant un bilan ou l'impact économique d'un foyer pour un pays ou alors quand l'information concernant des foyers est secondaire ;
- Non pertinent (général) pour les articles ne décrivant que des généralités des maladies modèles.

ANALYSE STATISTIQUE DU CORPUS

Le corpus étudié dans ce mémoire, via nos codes java, est constitué 1064 fichiers (notamment de 531 pairs de fichiers) dont 502 contiennent de données d'analyse via les champs « content », en revanche, 29 ne contiennent pas de données, leurs champs « content » ne se trouvent pas renseignés.

Tf (Term frequency) du corpus

Le but est d'obtenir la fréquence d'apparition de mot dans le document du corpus tout en renseignant le type ou la classe de ce document. Le résultat des calculs est présenté dans la figure 2.

```
9:zimbabw:349.json.processed.json.pos.racine:related
9:zealand:117.json.processed.json.pos.racine:unrelevant
9:wto:39.json.processed.json.pos.racine:related
9:winter:470.json.processed.json.pos.racine:unrelevant
9:wina:381.json.processed.json.pos.racine:related
9:websit:170.json.processed.json.pos.racine:unrelevant
9:wave:139.json.processed.json.pos.racine:new
9:virus:454.json.processed.json.pos.racine:new
9:virus:35.json.processed.json.pos.racine:new
9:virus:187.json.processed.json.pos.racine:related
9:virus:143.json.processed.json.pos.racine:new
9:vehicl:329.json.processed.json.pos.racine:related
9:vaccin:113.json.processed.json.pos.racine:related
9:usa:500.json.processed.json.pos.racine:related
9:usa:175.json.processed.json.pos.racine:unrelevant
```

Figure 2 : fréquence de termes dans chaque document du corpus

Dans le figure 2 ci-dessus, les colonnes sont séparées par le double point, la première colonne contient les fréquences de chaque mot, la deuxième correspond aux termes du vocabulaire de corpus, la troisième aux noms de fichiers respectifs, et la dernière au type ou classe du fichier.

Nous avons aussi calculé la fréquence de chaque mot dans l'ensemble de corpus.

```
half:31
directorate_assur:1
business_ent:1
gloomi:1
abrupt:1
hall:11
clinton:1
deadline_b:1
levison:1
cut_b:1
enclav:1
tef:1
stumbl:1
pick:8
smallholder_engag:1
bubble_fil:1
company_set:1
tel:1
health_hav:3
ten:9
system_be_voluntari:1
pretens:1
powder_hav:1
cdc:3
boar_b:8
expert_visit:1
```

Figure 3 : fréquence de termes dans l'ensemble du corpus

Df (Document frequency) de mots

Le but est de trouver pour chaque mot, le nombre de documents dans lesquels ce mot est présent. Par exemple, le mot *delphi* est présent dans 4 documents.

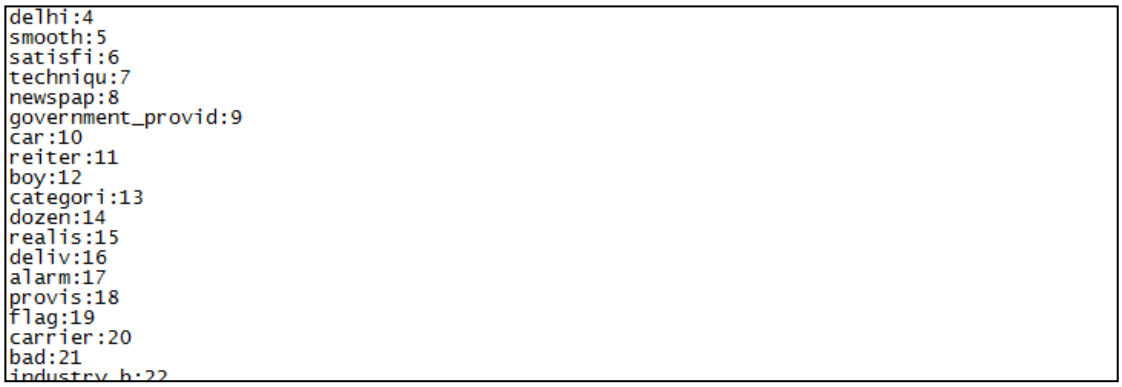

Figure 4 : df de mots

Le but est de trouver pour chaque mot, la fréquence de mot dans une catégorie ou classe en vue d'identifier son pouvoir de discrimination par rapport à une classe pour une meilleure possible réduction de l'espace de représentation.

Fréquence de mot dans une classe (Information mutuelle)

Nous avons calculé la fréquence de mots en fonction de la classe « new », « related » et « unrelevant », ainsi :

- Pour la classe new (dite aussi « pertinente », la plus importante), le résultat est présenté dans la figure 5.

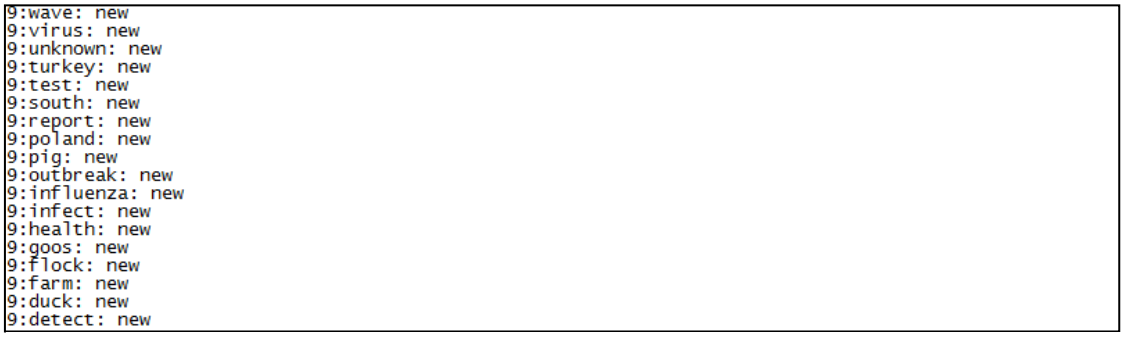

Figure 5 : la fréquence de mots dans la classe new

Pour la classe related :

- Pour la classe related (dite aussi «en relation». Par ex un bilan sur la maladie), le résultat est présenté dans la figure 6.

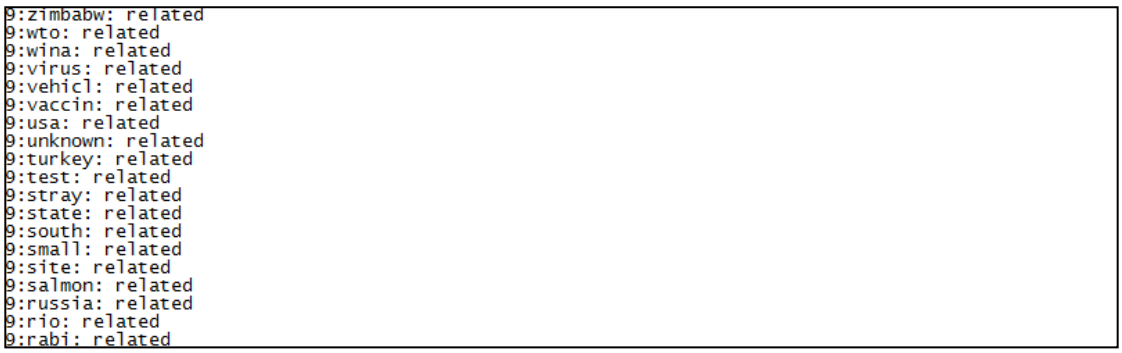

Figure 6 : la fréquence de mots dans la classe related

- Pour la classe unrelevant (dite aussi « non pertinente »), le résultat est présenté dans la figure 7.

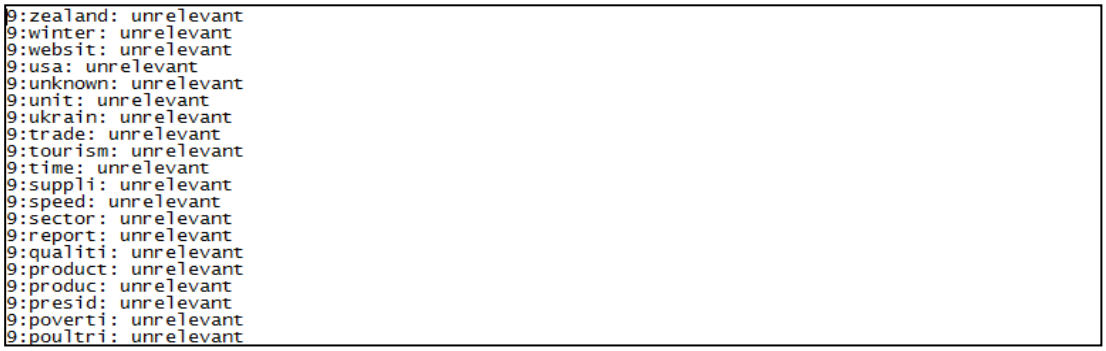

Figure 7 : la fréquence de mots dans la classe unrelevant

Pour certains traitements statistiques de corpus, nous utilisons, en raison de leur robustesse, des outils TAL d'exploitation de corpus déjà existant, à savoir :

- **AntConc** (développé par laurence Anthony) est un outil proposant différentes fonctionnalités d'exploration d'un corpus de texte, de la construction de la liste des mots d'un texte à la concordance, incuant une recherche fondée sur les expressions regulirère ;
- **Trameur** (développé par Serge fleury) est outil de représentation du texte en machine sous la forme d'une Trame et d'un Cadre, pour ensuite réaliser des calculs textométriques, les statistique et documentaire de textes en vue de leur profilage sémantique, thématique et de leur interprétation ;
- **Lexico** (a été conçu à l'ENS Fontenay-Saint-Cloud, au sein de l'équipe Lexicométrie et textes politiques dirigée par Maurice Tournier et par André Salem) est un outil de statistique textuelle, le corpus est d'abord découpé en parties et en mots, ensuite on applique les opérations.

Il est souhaitable, par ailleurs, de prendre en compte, dans le processus de classification, un maximum d'information sur les attributs ou descripteurs.

Différentes ressources linguistiques sont utilisées telles que les thésaurus, les ontologies. Ces ressources et leur exploitation en fouille de textes sont détaillées dans les sections suivantes.

RESSOURCES

Cette partie du manuscrit présente les ressources linguistiques lexico-sémantiques (ressources externes au corpus), la discussion des hiérarchies sémantiques (les structures, ou les vocabulaires plus ou moins contrôlés : la terminologie, le thesaurus, l'ontologie),et les dictionnaires, pour l'optimisation de processus de notre chaîne de traitement à l'étape de représentation de données textuelles que nous présenterons dans le prochain chapitre d'une part, et pour pallier notamment aux difficultés liées aux problèmes d'hétérogénéités de données (la synonymie, la polysémie, la désambiguïsation, etc.) d'autre part.

NOTIONS UTILES

Dans cette partie du manuscrit, nous présentons le pré-requis des structures lexico-sémantiques : mot, terme, concept, phénomène de la synonymie et de la polysémie.

Un mot

Un mot est une représentation (son ou groupe de sons articulés ou figurés graphiquement, constituant une unité de signification à laquelle est liée) d'un concept dans une langue donnée (http://www.cnrtl.fr/definition/mot).

Un terme

Un terme est un mot ou un groupe de mots représentant des traces linguistiques d'un concept (Mathieu, 2004). A cette définition s'ajoute : « un mot ou groupe ayant une signification spécifique dans un domaine donné »; Un **terme contrôlé** est un terme ayant une signification non ambiguë dont l'usage est recommandé ; Une **variante** est un terme différent ayant le même usage que le terme contrôlé.

Catégorie de termes étudiés

Nous distinguons deux catégories de termes : les descripteurs et les non-descripteurs.

- **Les descripteurs (ou les mots-clés)** sont les termes retenus et choisis parmi une collection de termes pour représenter, sans ambiguïté, un concept contenu dans un article de média ou stratégie de recherche documentaire.
- En revanche, **les non-descripteur**s sont des termes qui appartiennent aux lexiques mais qui ne sont pas adaptés pour la description des données textuelles ou la recherche documentaire.

Forme des descripteurs

Les différentes formes des descripteurs que l'on rencontre dans les travaux de fouille de textes sont les suivantes : la nature, le genre, le nombre grammatical, l'orthographe, le nom propre, etc.

Un concept

Un concept est une représentation, ou une construction abstraite et formelle.

Synonymie et polysémie /les contraintes

Plusieurs termes peuvent désigner un même concept, par exemple [lapaka] et [lopopo] sont deux termes qui se réfèrent au fruit pastèque (phénomène de synonymie) ou qu'un même terme peut désigner différents concepts (phénomène de polysémie). La figure 8 illustre bien un exemple de synonymie.

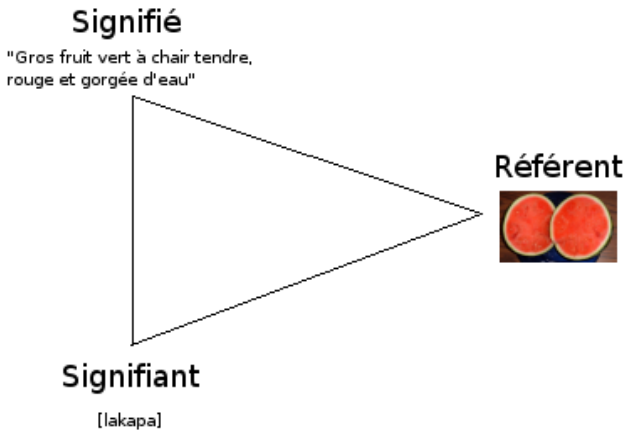

Figure 8 : Double articulation du langage avec Andrée martinée, pour illustrer deux termes se référant à un fruit (le concept pastèque). Source : https://fr.wikibooks.org/wiki/La_morphologie

Dans la littérature, de nombreuses propositions ont été formulées pour mieux rendre compte de ces deux phénomènes.

Beaucoup de ces propositions reposent sur l'utilisation de ressources lexico-sémantiques telles que les dictionnaires, thésaurus ou ontologie.

Toutefois ce type d'approche présente deux inconvénients majeurs : tout d'abord, le développement de telles ressources est extrêmement coûteux ; ensuite, une langue est loin d'être figée.

En effet, les ressources linguistiques lexico-sémantique dont on peut disposer à un instant donnée ne seront que partielles et ne fournissent pas une solution complète aux problèmes de synonymie et polysémie (Amini and Gaussier, 2013).

De plus, des nombreux travaux, dans la littérature, ont montré que l'usage des ressources sémantiques n'apporte pas beaucoup d'information, la représentation vectorielle dans les systèmes de traitements automatiques permet d'atteindre une meilleure performance pour différentes tâches de fouille de textes (la classification dans notre cas).

LA TERMINOLOGIE

Définitions

En 2000, Philippe Lefèvre, définit : « Les terminologies sont des listes de termes d'un domaine ou sujet donné représentant les concepts ou notion les plus fréquemment utilisés ou les plus caractéristiques, cette liste étant ou non structurée ».

La terminologie s'appuie sur trois principales notions suivantes :

1. Le référent (la chose ou l'objet du monde) ;
2. Le signifiant (la forme ou le signe) ;
3. Le signifié(le concept ou l'idée que dénote le signe).

La terminologie consiste en l'étude du choix et de l'usage des termes faisant partie des vocabulaires spécialisés ou des domaines

Vocabulaire de domaine /Termes contrôlés

Un vocabulaire est une collection de termes permettant de désigner le plus clairement possible les sujets traités selon le contenu des données textuelles.

Dans ce travail, un certain nombre de mots-clés ont été proposés, préalablement, dans l'étape d'acquisition automatique de données de notre corpus d'étude. Une construction de la terminologie spécialisée a donc été réalisée par une équipe de spécialistes en épidémiologie animale au sein de l'unité mixte de recherche (ASTRE) pour des recherches automatiques spécialisées d'articles de média sur le web. Le tableau illustre une partie de la liste de termes spécialisés ou contrôlés.

Outbreak	outbreaks	Case	cases
contamination	Loss	Disease	infection
Infestation	pathogen	Virus	bacteria
Illness	syndrome	Introduce	incursion
identification	Spread	Diffusion	emergence
Discovery	detection	Restriction	…

Tableau 1 : exemples de termes (en anglais) relatifs aux signes cliniques pour assurer une veille syndromique

Le principal inconvénient d'une terminologie (dans notre cas une liste de mots-clés, voir le tableau ci-dessus) dans la littérature est son niveau de structuration assez limité.

Les relations supportées entre termes dans la structure lexico-sémantique sont purement linguistiques (synonymie, hyponymie, etc.).

Le pouvoir d'expression de relations entre termes étant faible, on lui préfère l'usage d'un thésaurus.

LE THESAURUS

Un thésaurus est un vocabulaire contrôlé. Il regroupe une collection de termes structurés choisis pour leur capacité à décrire un domaine.

En 1987, l'AFNOR propose la définition suivante : « Un thésaurus est une structuration hiérarchisée d'un ou plusieurs domaines de la connaissance et dans les notions sont présentées par des termes de la langue naturelle et les relations entre notions par des signes conventionnels ».

Et GDT (Grand dictionnaire terminologique) de l'office québécois de la langue française (OQLF) définit un thésaurus comme étant : « un vocabulaire contrôlé et dynamique de termes ayant entre eux des relations sémantiques et génériques, et qui s'applique à un domaine particulier de la connaissance ».

L'objectif d'un thésaurus est de constituer un vocabulaire normalisé et d'organiser la liste des termes de ce vocabulaire sans forcément les définir.

On différencie au moins deux types de termes reliés classiquement au sein du thésaurus par trois types de relations :

- Les termes descripteurs sont reliés aux termes synonymes non descripteurs par des relations d'équivalence ;
- Les termes descripteurs sont organisés en une hiérarchie par une relation entre terme générique et terme spécifique ;
- Les termes descripteurs sont reliés en champs thématiques, les sujets connexes et termes associés par une relation d'association.

En pratique, un thésaurus fournit une structure pour le classement, l'analyse et indexation de d'articles.

La principale insuffisance de thésaurus est qu'il est difficilement utilisable de représenter les relations propres à un domaine.

L'introduction des notions telles que les classes et les propriétés, fait que l'ontologie dispose d'un pouvoir expressif plus riche (Teguiak, 2012).

L'ONTOLOGIE

Le terme ontologie fut emprunté, à la philosophie au début des années 1990, par l'informatique, où il signifie science ou théorie de l'être, une branche qui étudie l'être en tant qu'être, l'existence en général (Gandon et al., 2012).

Dans le sous-domaine de l'intelligence artificielle qui s'intéresse à la représentation des connaissances, la définition est différente.

Dans la littérature, la définition la plus citée, présente une ontologie comme étant : «une spécialisation explicite de conceptualisation (la représentation du monde par rapport à un domaine donné sous forme d'un ensemble de concepts tels que classes, individus, attributs, leurs définitions et leurs interrelations) partagée» (Gruber, 1993).

A cette définition s'ajoute une légère modification formulée ainsi : « une ontologie est une spécialisation explicite (signifie que le type des concepts et les contraintes sur leurs utilisations sont explicitement définis) et formelle (se réfère au fait que la spécialisation doit être lisible) d'une conceptualisation partagée (se rapporte à la notion selon laquelle une ontologie capture la connaissance consensuelle) (Borst and Borst, 1997)».

En pratique (dans la machine), une ontologie informatique est un objet logiciel, un ensemble structuré des termes et concepts représentant le sens d'un champ d'information, un modèle de données représentatif d'un ensemble de concepts dans un domaine, une représentation de propriétés générales de ce qui existe pour une application dans un formalisme supportant un traitement rationnel (Gandon et al., 2012).

Analyse des approches de représentations textuelles pour la catégorisation de textes

Les tâches de fouille de textes, en particulier la tâche de classification de textes que nous mettons en œuvre, nécessite une première étape, la représentation textuelle.

Deux grandes approches s'offraient, pour la catégorisation de textes : la représentation vectorielle basée sur la terminologie du corpus (dite aussi représentation par sac de mots avec « racines lexicales (en anglais stemms) », ou « lemmes », « n-grammes ») sans l'utilisation de ressources sémantique, et la représentation vectorielle fondée sur les ressources externes lexico-sémantiques (dite aussi « représentation conceptuelle »).

Dans ce travail, nous avons principalement employé l'approche vectorielle ou sac de mots basé sur la terminologie de textes du corpus d'entrainement, ensuite combiné avec la terminologie contrôlée ou une liste de mots-clés élaboré par les experts du domaine (de l'unité ASTRE, pour notre cas), pour la catégorisation de textes.

LA REPRESENTATION VECTORIELLE (APPROCHE SAC DE MOTS BASEE SUR LA TERMINOLOGIE DU CORPUS)

Cette partie de notre étude décrit le processus d'extraction des descripteurs depuis notre corpus, d'effectuer différentes manipulations sur ces données (prétraitement), de choisir automatiquement les descripteurs à partir de ces données et de calculer leurs pondérations, enfin de produire une matrice de termes pondérés.

1. ANALYSE ET REPRESENTATION LINGUISTIQUE

Dans le cadre du processus de traitements de textes, il est rare de travailler sur des corpus bruts. Ceux-ci sont généralement soumis à une phase de prétraitement ou de normalisation.

Durant cette phase de notre approche, les textes subissent des modifications, des opérations qui facilitent les traitements ultérieurs.

1.1. Constitution des données textuelles à traiter

Nous ne saurons réaliser les opérations sans mettre à dispositions de notre programme les données. Ainsi cette partie permet l'extraction de données à partir d'une collection (corpus) constituée de paires de fichiers dans lesquels le texte de chaque document est rangé dans le champ « content » respectif du fichier en format « json », et « l'identifiant et l'information » sur la classe de chaque document sont placé dans le fichier d'annotation.

Cette partie vise donc à mettre à disposition du programme, les textes du champ « content » de fichier json, pour nos traitements de classification. Le processus d'extraction est illustré en Figure 9 et 10.

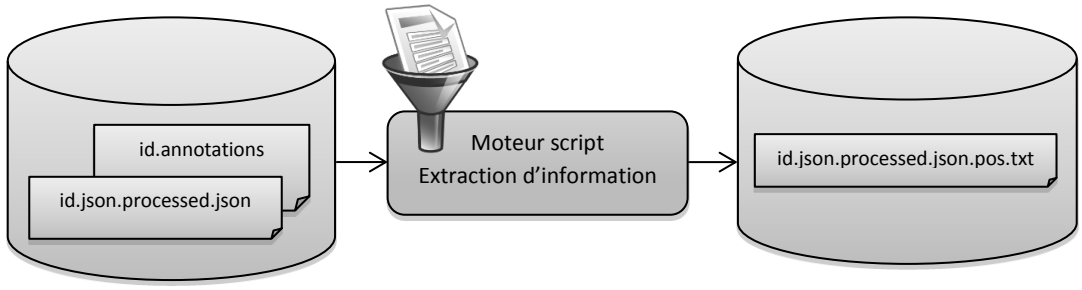

Figure 9 : Processus d'extraction d'information du corpus

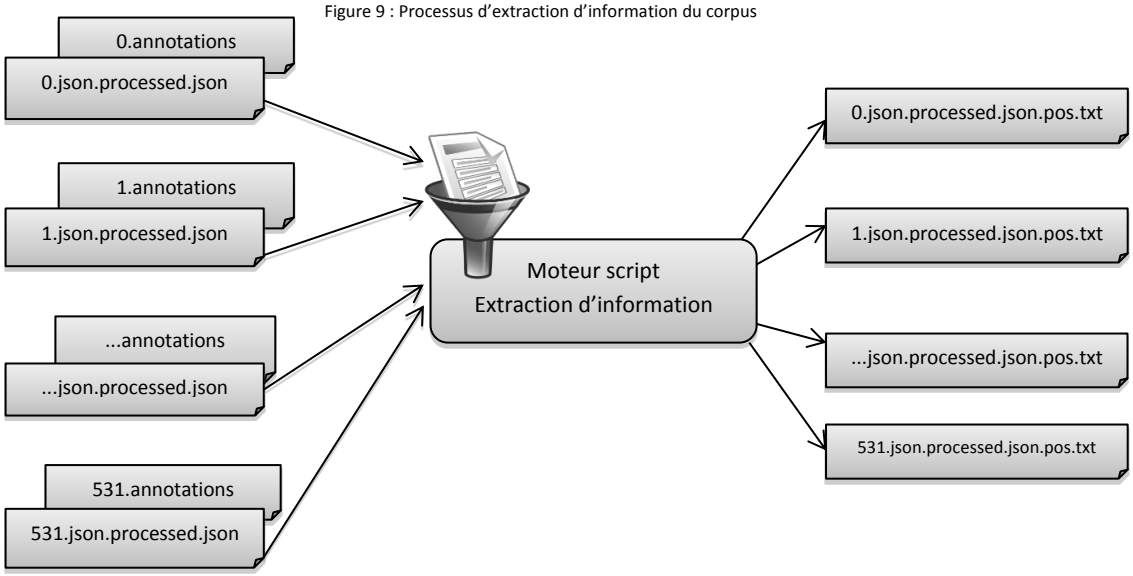

Figure 10 : Processus détaillé d'extraction d'information du corpus.
A chaque paire de fichiers en entrée, correspond un fichier de texte en sortie, aprèsexécution de notre programme ou module d'extraction de données.

1.2. Analyse et étiquetage grammatical de texte

Lors de cette étape, les textes du corpus ont été segmentés, étiquetés et lemmatisés. Nous avons utilisé l'outil TreeTagger pour analyser et associer automatiquement des étiquettes de rôle grammatical sur des sorties du script d'extraction d'informations textuelles de l'étape précédente.

Nous avons donc recupéré en entrée le corpus en format txt contenant les données extraites à partir du script d'extraction, pour avoir une sortie de fichiers treetagger.

Nous signalons que l'intrégration dans nos scripts de l'outil treetagger a occasionné l'écriture de scripts bash, ce qui a pour conséquence l'utilisation massive de fichiers bash pour l'automatisation des opérations respectives possibles des appels de fonctions systèmes afin d'étiqueter les textes.

Afin d'éviter les écritures de scripts bash, nous avont utilisé l'api tt4j dans notre progamme pour étiqueter les textes de sortie du module d'extraction. Un exemple de sortie de notre script en bash ou tt4j et treeatagger est présenté en Figure 11.

```
State       NP      State
Government          NP      Government
said        VVD     say
on          IN      on
Tuesday     NP      Tuesday
that        IN/that that
it          PP      it
had         VHD     have
killed      VVN     kill
hundreds            NNS     hundred
of          IN      of
fowls       NNS     fowl
infected            VVN     infect
by          IN      by
the         DT      the
```

Figure 11 : Fichier de sortie treetagger

La première colonne contient le mot ou token, la deuxième la catégorie ou fonction grammaticale, et la troisième le lemme ou forme canonique. Lorsque le mot n'est pas reconnu par treetagger, selon le paramétrage, l'étiquette <unknown> est donnée.

Treetagger nous donne cependant la possibilité d'optimiser le traitement, comme par exemple, d'étendre son dictionnaire de la reconnaissance du vocabulaire. Deux choix s'offraient : l'extraction de la colonne lemme tout en ignorant les mots non reconnus via le motif <unknown>, ou la possibilité de travailler avec le motif <unknown> pour récupérer les mots non reconnus par treetagger. Nous avons préféré le premier choix.

Afin de faciliter les opérations ultérieures des filtrages grammaticaux, à partir des fichiers de format txt produits lors du module d'extraction, notre script d'étiquetage a aussi produit des sorties xml dont un exemple est présenté en figure 12.

```
<element>
 <data type="type">dt</data>
 <data type="lemma">the</data>
 <data type="string">the</data>
</element>
<element>
 <data type="type">np</data>
 <data type="lemma">rivers</data>
 <data type="string">rivers</data>
</element>
<element>
```

Figure 12 : Fichier de sortie xml ou de transformation de sorties treetagger. L'attribut type de la balise data représente respectivement les trois colonnes de la structure de fichier sortie treetagger, dans le document xml.

L'ensemble des documents en format xml produits par notre programme, occupent d'espace disque par rapport au format de sortie txt de treetagger. En effet, pour un étiquetage d'une collection (de 503 fichiers de textes) de 1,29 Mo, nous avons obtenu un dossier de 27,5Mo de l'ensemble de textes étiquetés en treetagger et balisé en xml, le même dossier en format original de textes étiqueté en trretagger en txt et sans balise xml permet d'obtenir une taille de 1,5Mo.

Nous avons préféré travailler sur les fichiers de sortie treetagger dans un format txt, en entrée de notre module de filtrage grammatical (les règles de combinaisons seront détaillées dans la section suivante) afin de réaliser la sélection de termes sur les colonnes lemmes de chaque fichier du corpus d'entrainement.

Un exemple de sortie de fichier contenant la colonne lemme de sorties treetagger obtenue à partir de notre script, avant le filtrage grammatical est présenté en figure 13.

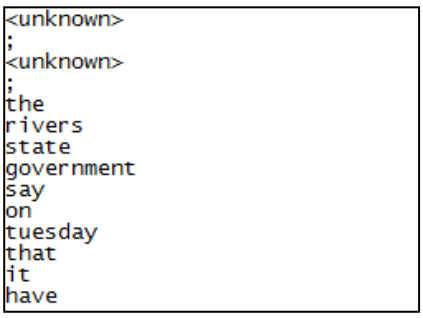

Figure 13 : Fichier de sortie extraction lemme de treetagger

Nous avons également testé l'étiqueteur de la librairie d'apache opennlp et celui de Stanford, ils s'intègrent bien dans notre script. Les opérations d'étiquetage facilitent, ainsi, la réalisation de filtrage des données (la première étape de réduction de la taille de vocabulaire), pour ne garder que les types de mots les plus susceptibles d'être porteurs d'informations de la thématique.

2. FILTRAGE DE DONNEES TEXTUELLES

Dans cette partie de notre approche, nous présentons différents techniques de filtrages qui constituent la première étape de réduction de la taille de vocabulaire (pour la robustesse de notre approche) dans l'espace de représentation.

Ainsi, à partir des fichiers de sorties treetagger, nous avons pu extraire des formes linguistiques (les termes depuis la colonne lemme de chaque fichier) grâce à notre programme.

2.1. Filtrage grammatical dynamique

Lors de cette étape de filtrage (linguistique), nous avons recouru à un certain nombre de règles afin de récupérer les mots grammaticalement importants pour notre espace de représentation de données pour la classification.

Nous avons laissé, via notre programme, la possibilité de modifier dynamiquement les paramètres, afin de tester respectivement l'efficacité de chaque combinaison de règles grammaticales.

Pour filtrer les mots grammaticalement importants (nom, verbe, adjectif, de mots pleins), nous avons utilisé la combinaison des règles suivantes :

- les noms (noms communs et noms propres)
- les verbes
- les adjectifs
- noms + verbes
- noms + verbes + adjectifs

Pour appliquer ces règles, nous avons eu en entrée, les fichiers des sorties treetagger filtré par colonne de lemme, et en sortie le dossier des fichiers contenant des formes issues de nos combinaisons de règles. Un exemple de fichier de sortie des opérations de filtrage grammatical est présenté en figure 14.

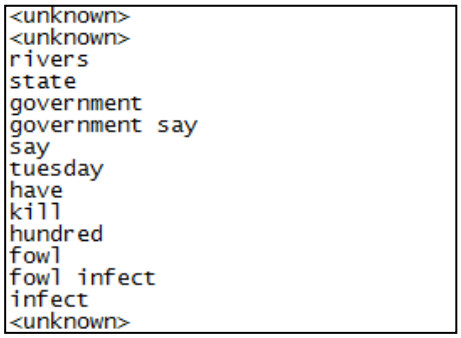

Figure 14 : Fichier de sortie de filtrage grammatical

2.2. Filtrage par liste de mots vides (l'anti-dictionnaire)

A partir d'une collection des fichiers textes produits par le filtrage grammatical, nous avons appliqué des transformations linguistiques telles que la lemmatisation ou la racinisation, et des opérations de suppressions des mots dont la taille de la chaîne est inférieure à un seuil, pour notre cas nous avons choisi 2, afin d'éliminer, automatiquement de l'espace de représentation, des formes telles que des ponctuations, des caractères spéciaux, la normalisation linguistique ou surfacique.

Normalisation de mots

La normalisation de mots est un processus qui transforme tous les mots d'une même famille sous une forme normale (ou canonique) de façon à ce que l'appariement ou la correspondance puisse avoir lieu entre les termes en relation d'équivalence.

Deux types de normalisation sont identifiées (Amini and Gaussier, 2013) :

- La normalisation surfacique (dite aussi « textuelle »), effectue quelques transformations superficielles telles que les ponctuations, les accents, la casse, les dates, les valeurs monétaires, sur les mots ou séquences de caractères.

- La normalisation linguistique consiste à ramener les mots fléchis à leurs formes canoniques.

  Deux types de normalisation linguistique sont identifiés :

  - La racinisation ;
  - La lemmatisation.

Afin supprimer les caractères spéciaux accolés au début de mot, d'appliquer la normalisation surfacique dans nos texte, nous avons utilisé, dans notre programme, des expressions régulières, tout en respectant les formes de mots composés ou poly-lexicaux, les dates, les valeurs monétaires nous ne les avons pas traité.

Concernant la lemmatisation ou racinisation, nous avons utilisé l'algorithme de Porter et Namer, avec une approche par dictionnaire intégré via le projet ou la librairie snowball.

Un exemple de fichier de sortie, après traitement des opérations linguistiques, est présenté en figure 15.

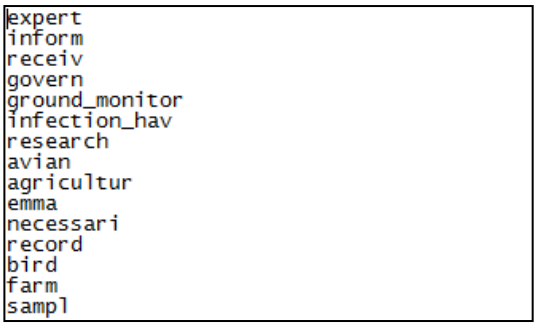

Figure 15 : Fichier de sortie de filtrage et normalisation linguistique

Anti-dictionnaires

L'anti-dictionnaire ou la liste de mots ou d'expression, est une liste de mots (vide de sens, sans intérêt, qui n'apportent donc pas d'informations) qui tendent à être présent avec une fréquence très élevée dans tous les documents d'une collection.

Dans notre approche, les mots vides, la forme <unknown>, sont supprimé par notre programme. Egalement, les mots ou formes dont la taille est inférieure à 2, telles que les ponctuations, les caractères spéciaux isolés, les caractères spéciaux en début de chaque mot, détectés automatiquement grâce à des motifs d'expressions régulière, sont automatiquement et respectivement supprimés dans les fichiers de sortie de notre programme ou script de normalisation.

En outre, nous avons recouru à l'utilisation d'une liste ou un fichier de mots vides qui nous a permis de filtrer des mots dans l'espace de représentation, par l'approche anti-dictionnaire. La liste de mots vides mise à œuvre est présentée dans le tableau 2.

a	a's	able	about	above	according	accordingly	across
actually	after	afterwards	again	against	Ain	ain't	all
allow	allows	almost	alone	along	already	also	although
always	am	among	amongst	an	and	another	any
anybody	anyhow	anyone	anything	anyway	anyways	anywhere	apart
appear	appreciate	appropriate	are	aren	aren't	around	as
aside	ask	asking	associated	at	available	away	Awfully
b	be	became	because	become	…		

Tableau 2 : Liste de mots vides

Dans la littérature, l'anti-dictionnaire peut jouer le rôle de dictionnaire, dans le cas, par exemple, d'une étude sur la fréquence d'apparition des mots vides.

Nous avons alors, après ces différents traitements linguistiques, des nouvelles formes lexicales sur lesquelles appliquer le filtrage statistique.

2.3. Filtrage statistique

Nous éliminons dans la base, lors de cette étape, les mots trop rares, les hapax, les mots trop fréquents et dont les pouvoirs de discrimination sont trop faibles pour la classification. En entrée de notre programme, nous travaillons sur le dossier des fichiers contenant des termes issus de transformations linguistiques décrites dans la section précédente, et en sortie, notre programme renvoie une liste de mots pertinents (descripteurs) représentant les attributs de notre matrice « sac de mots » selon différentes pondérations (nombre d'occurrences, booléen, tfidf) pour la tâche de classification.

Afin d'éliminer les mots trop fréquents dans notre espace de représentation, nous avons appliqué un processus d'élagage avec un seuil fixé (appelé *seuillage*). Les termes n'obéissant pas à ce seuil, c'est-à-dire, les termes qui ne sont pas présents un nombre de fois donné dans le corpus sont éliminés dans notre base ou liste d'attributs. Les termes qui sont conservés dans la base sont ceux qui respectent le seuil donné. Dans nos expérimentations, nous nous sommes appuyés sur un seuil propre à la présence du descripteur au moins une fois, deux fois, trois fois, quatre fois ou cinq fois dans le corpus.

3. MATRICE DE POIDS

Lors de cette partie, nous réduisons les textes de notre corpus à l'état de sacs de mots. Ensuite, nous construisons, une matrice de poids pour les opérations de classification.

Cette matrice de poids de termes implique un choix ou une sélection de variables ou d'attributs constituant les descripteurs linguistiques des textes ou documents, et la pondération des termes à travers des calculs d'occurrences, de fréquences ou d'apparitions des termes pour caractériser le document. L'idée sous-jacente est que les termes importants doivent avoir un poids fort. Le squelette de cette matrice est présenté dans le tableau 3.

	Teme1	Terme2	Terme3	…	Terme n
Document 1	Poids 11	Poids 12	Poids 13	…	Poids 1n
Document 2	Poids 21	Poids 22	Poids 23	…	Poids 2n
Document 3	Poids 31	Poids 32	Poids 33	…	Poids 3n
Document 4	Poids 41	Poids 42	Poids 43	…	Poids 4n
Document 5	Poids 51	Poids 52	Poids 53	…	Poids 5n
…	…	…	…	…	…
Document n	Poids n1	Poids n2	Poids n3	…	Poids nn

Tableau 3 : squelette de notre matrice de valeurs ou poids

Dans cette espace de représentation matricielle, les lignes représentent les documents, les colonnes attributs ou termes, à l'intersection des lignes et des colonnes les valeurs booléennes, ou les poids tf ou les scores tf.idf.

4. SELECTION DE DESCRIPTEURS OU D'ATTRIBUTS

Cette étape concerne le choix d'attributs (termes pertinents, descripteurs linguistiques) les plus discriminants d'un document.

La sélection de descripteurs est un domaine très actif depuis de nombreuses années. Il permet d'obtenir un sous-ensemble d'attributs à partir d'un ensemble original, de sorte que la taille du vocabulaire soit réduite de façon optimale dans l'espace de représentation (la matrice attributs-articles) tout en essayant de maintenir ou d'améliorer les performances d'algorithme de classification.

4.1. Notre approche de sélection de sous-ensemble d'attributs

Pour rappel, Liu et Yu en 2005, ont les trois axes d'identifié suivant : les stratégies ou méthodes de recherche (séquentielle, aléatoire, et complète), les critères d'évaluation (Filter, Wrapper) et les algorithmes de classification (SVM, Naïve bayes, etc.).

Après différentes opérations réalisées préalablement par notre programme sur le corpus d'étude, entre autres, le filtre grammatical, le filtre par liste de mots vides (l'usage d'anti-dictionnaire), la normalisation surfacique, le filtre statistique, l'élagage, la taille de vocabulaire est toujours élevée dans notre espace de représentation variables ou d'attributs.

En effet, à partir de notre corpus, le vocabulaire représentant nos descripteurs ou attribut sa une taille importante (autour de 2700 termes).

Afin de réduire la taille du vocabulaire, la démarche proposée consiste à utiliser parmi les approches qui existent (flitre, etc.), le critère d'évaluation dite Wrapper qui consiste à l'évaluation des attributs. Ce processus s'interface avec SVM et la stratégie ou méthode de recherche BestFirst, pour produire un sous-ensemble de termes les plus discriminants.

Nous avons également testé pour la méthode ou stratégie de recherche Ranker, les critères d'évaluation suivants : le gain d'information (InfoGainAttributeEval), l'Analyse de Composante Principale (dite ACP, PrincipalComponents), le Relief (ReliefAttributeEval), l'information mutuelle (CorrelationAttributeEval).

Nous avons opté dans notre cas pour une démarche qui utilise le Wrapper avec BestFirst et SVM. Ainsi chaque article soumis à notre modèle de fonction sera alors décrit par un sous-ensemble d'attribut issu de notre approche fondée sur la stratégie de recherche BestFirst et de critère dite Wrapper.

5. PONDERATION DES DESCRIPTEURS

A partir du vocabulaire de notre corpus filtré, nous effectuons des calculs respectifs d'occurrences, de fréquences ou d'apparitions de termes du corpus afin d'obtenir une matrice de pondérations pour la tâche de classification.

En effet, plusieurs stratégies de calculs de pondérations sont possibles. Ces dernières sont décrites dans les sous-sections suivantes.

5.1. La fréquence de termes

La mesure de la fréquence d'un terme est le nombre d'occurrences de ce terme, c'est un bon indicateur de l'importance de ce terme dans un document considéré, plus sa valeur ou son score est élevé, plus le terme est important dans la description du document considéré.

Afin de remplir la matrice du nombre d'occurrences, pour la tâche de classification, nous avons automatisé le calcul tf afin de pondérer chaque terme (ou attribut) et d'assigner à l'intersection de ligne correspondant au document (ou article) et de colonne correspondant aux variables (attributs ou termes pertinents). Ainsi, en entrée de notre programme de calculs d'occurrences de termes, nous considérons un ensemble d'articles traités, le corpus filtré, et un jeu de variables, attributs ou termes pertinents, et en sortie, une matrice de valeurs d'occurrences de chaque terme du vocabulaire de notre corpus filtré.

Dans notre programme de classification, un module se charge de produire, à partir d'une matrice de valeurs d'occurrence, une sortie weka au format arff, en vue d'une expérimentation pour guider notre choix du modèle de classification. Un exemple de fichier de sortie weka au format arff est présenté en figure 16.

```
@attribute arabia_b real
@attribute categor real
@attribute delhi real
@attribute rival real
@attribute efficaci real
@attribute s_class {new,related,unrelevant}
%The Data of the ARFF file looks like the following:
@data
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
101,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
102,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
103,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
104,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,0,0
105,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0
```

Figure 16 : Fichier de sortie weka à partir d'une matrice d'occurrences. La première colonne après @data, contient les identifiant de nos différents documents.

5.2. L'approche boolénne

L'approche booléenne est basée sur la présence ou l'absence d'un terme dans un document. La mesure de poids binaire d'un terme est 1 si ce terme est présent dans le document, 0 sinon. Il existe un module, depuis notre programme, qui se charge de calculer le poids

binaire de chaque terme, et produit en sortie une matrice de valeurs ou poids binaire. A partir de ce tableau, un autre module de notre programme se charge de produire une sortie weka au format arff, en vue d'une expérimentation ultérieure dans le but de guider le choix de classification. Un exemple de fichier de sortie weka est présenté en figure 17.

```
@attribute arabia_b real
@attribute categor real
@attribute delhi real
@attribute rival real
@attribute efficaci real
@attribute s_class {new,related,unrelevant}
%The Data of the ARFF file looks like the following:
@data
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0
10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
101,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
102,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
103,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
104,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0
105,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
106,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
107,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
108,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
109,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 17 : Fichier de sortie weka à partir d'une matrice de poids booléenne

5.3. L'approche tf.idf

La mesure tf.idf (term frequency inverse document frequency) est le poids (final) d'un terme dans un espace de représentation. Elle combine deux mesures (ou poids initiaux), le tf et l'idf, pour rendre compte du pouvoir discriminatoire d'un terme.

Elle augmente avec la fréquence du terme dans le document (on fait référence à la mesure tf) et sa rareté dans le corpus ou collection de textes (on parle de la mesure idf). Un module de notre programme réalise le calcul de poids tf.idf à partir d'un jeu de termes pertinents et d'une collection de documents. Il produit en sortie une matrice de valeurs tf.idf.

Un autre module se charge ensuite de produire un fichier arff en vue d'une expérimentation dans weka pour le choix d'un modèle pertinent pour notre classifieur. Un exemple de fichier de sortie weka pour représenter la matrice de poids tf.idf est présenté en figure 18.

```
@attribute categor real
@attribute delhi real
@attribute rival real
@attribute efficaci real
@attribute s_class {new,related,unrelevant}
%The Data of the ARFF file looks like the following:
@data
1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
10,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.075044600053
100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.05419887781
101,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
102,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
103,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.026
104,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
105,0.018232517093382895,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
106,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
107,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
108,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
109,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.014
```

Figure 18 : Fichier de sortie weka à partir d'une matrice de poids TfIdf

1.3. Evaluation d'attributs de la matrice.

Cette partie du manuscrit est dédié à la sélection ou la réduction de la taille du vocabulaire de notre corpus pour garantir une certaine robustesse de la fonction de prédiction.

Elle donne un aperçu des résultats issus des opérations de filtrage et des sélections de descripteurs.

Sur le plan quantitatif

Avant l'élagage nous avons une matrice de 11409 (en raison de 11409 descripteurs ou termes de notre vocabulaire). Ainsi le tableau 4, donne les résultats de seuillage sur la fréquence des termes, c'est-à-dire le seuil sur le nombre de documents dans lesquels le terme est présent :

Seuil, au moins	Nombre de descripteurs selon l'élagage
1	5410
2	3899
3	3326
4	2975
5	2754

Tableau 4 : Evaluation matrice de poids (seuillage, élagage)

Sur le plan qualitatif

La sélection d'un sous ensemble d'attributs ou termes, par la méthode d'évaluation individuelle de variable, l'approche wrapper, nous a permis de réduire l'espace de représentation matricielle de termes de façon optimale et d'améliorer les performances. La taille du vocabulaire est donc passée de 2754 à 370 termes.

LA REPRESENTATION VECTORIELLE FONDEE SUR LES RESSOURCES LEXICO-SEMANTIQUES

Cette étape de notre travail présente le modèle de représentation textuelle qui s'appuie sur des ressources sémantiques (représentation conceptuelle). Ces dernières reposent sur une liste de termes contrôlés proposée par des experts du domaine.

REPRESENTATION CONCEPTUELLE (VECTEUR SEMANTIQUE)

Le vecteur sémantique est un modèle qui a été introduit par Chauché en 1990. Il permet l'utilisation de thésaurus comme base d'espace vectoriel de documents.

Ainsi dans la littérature, la représentation conceptuelle est une approche qui s'appuie sur des concepts des ressources lexico-sémantiques pour découvrir les correspondantes entre les concepts et leurs les variantes (les termes) durant la construction d'espace de représentation vectorielle.

Certains travaux s'appuient sur des ressources telles que WordNet, pour par exemple mapper des mots en synsets (syn pour synonyme et set pour ensemble) afin de pouvoir construire une représentation conceptuelle pour la catégorisation (multilingue) des textes. La figure 19, illustre bien la représentation conceptuelle de documents.

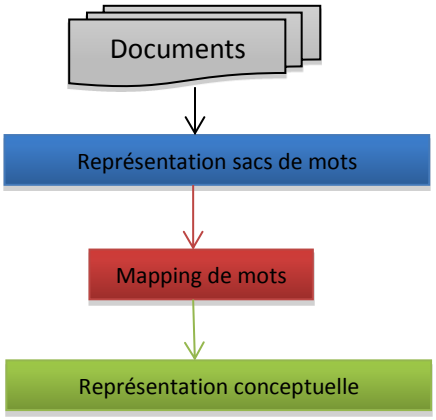

Figure 19 : représentation conceptuelle de textes

REPRESENTATION VECTORIELLE FONDEE SUR LA TERMINOLOGIE CONTROLEE (NOTRE APPROCHE)

Cette approche originale consiste à modifier ou à ajuster, dans l'espace de représentation, les poids de chaque article en fonction de la présence de chaque terme issu du vocabulaire contrôlé construit par expert du domaine. En d'autres, nous allons pondérer fortement les termes du domaine préalablement identifiés par des experts du domaine. La figure 20, illustre bien la représentation vectorielle fondée sur la terminologie.

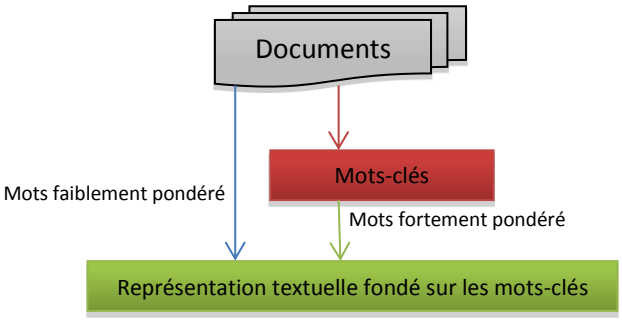

Figure 20 : Représentation vectorielle fondé sur la terminologie

Classification automatique

Cette partie de notre étude concerne l'évaluation des différents modèles afin de choisir les paramètres les plus adaptés pour identifier le« meilleur » modèle pour la classification de document. Ce modèle sera alors intégré à la plateforme Padiweb (cf. section suivante).

SELECTION D'ALGORITHME DE CLASSIFICATION (CLASSIFIEUR)

Pour nos expériences, nous avons choisi de travailler sur les trois modèles les plus répandus dans la littérature, à savoir SVM (ou MVS en français, Machine à Vecteur de Support), le Naïve Bayes et J48 (arbre de décision).

ENTRAINEMENT ET SAUVEGARDE DE MODELE

Entre le découpage des données en deux grands ensembles (entrainement et test) et le partitionnement en n-strate (validation croisée), nous avons préféré choisir la seconde approche. Elle permet de tester tous les exemples (de l'ensemble d'apprentissage) dans la précision prédictive (de modèle) de fonction même lorsque la population à étudier est trop petite. L'évaluation prédictive de notre modèle est alors basée sur la validation croisée, ensuite sauvegarder pour la catégorisation des textes dans la phase prédictive de classes.

EXECUTION D'ALGORITHME

Lors de cette étape, nous avons testé respectivement les trois algorithmes afin de choisir le meilleur d'entre eux. Le plan d'exécution est présenté en figure 21.

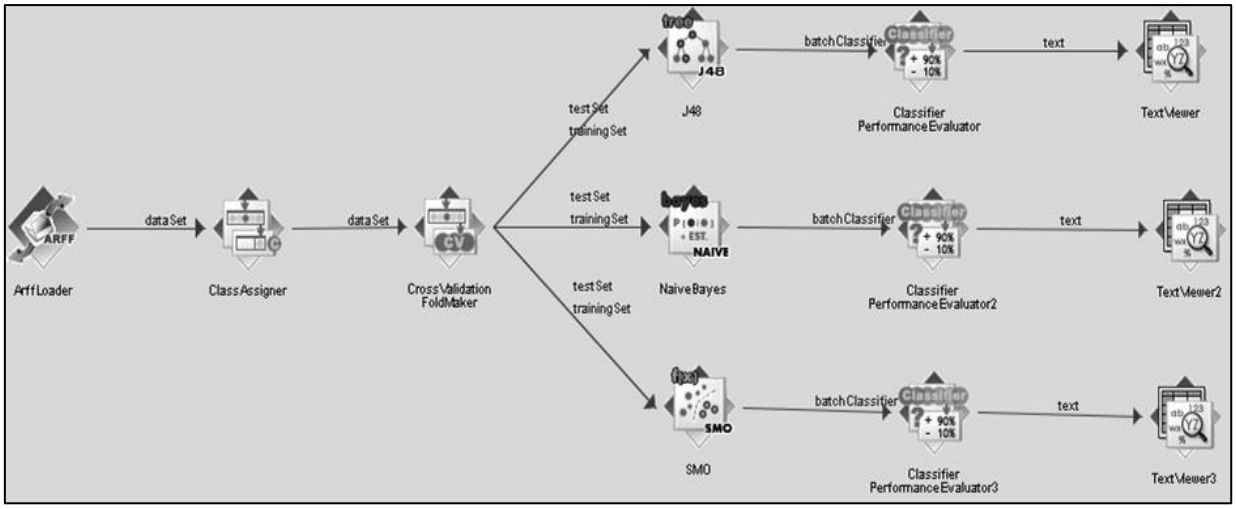

Figure 21 : Le plan d'exécution d'évaluation des modèles sous Weka

Dans le workflow (Figure 21) nous avons six étapes :

1. Le chargement du fichier arff ;
2. L'assignation des classes ;

3. La spécification de jeu de données pour l'entrainement et test, par le mécanisme de cross validation (la validation croisée) afin de subdiviser notre jeu de données en 10 strates (9 pour l'entrainement et 1 pour le test, respectivement pour chaque strate) ;
4. Test d'algorithmes J48, Naïve Bayes et SVM ;
5. Evaluation de la performance des algorithmes ;
6. Visualisation des résultats dans une zone de texte.

Les résultats du classifieur sont présentés en figure 22.

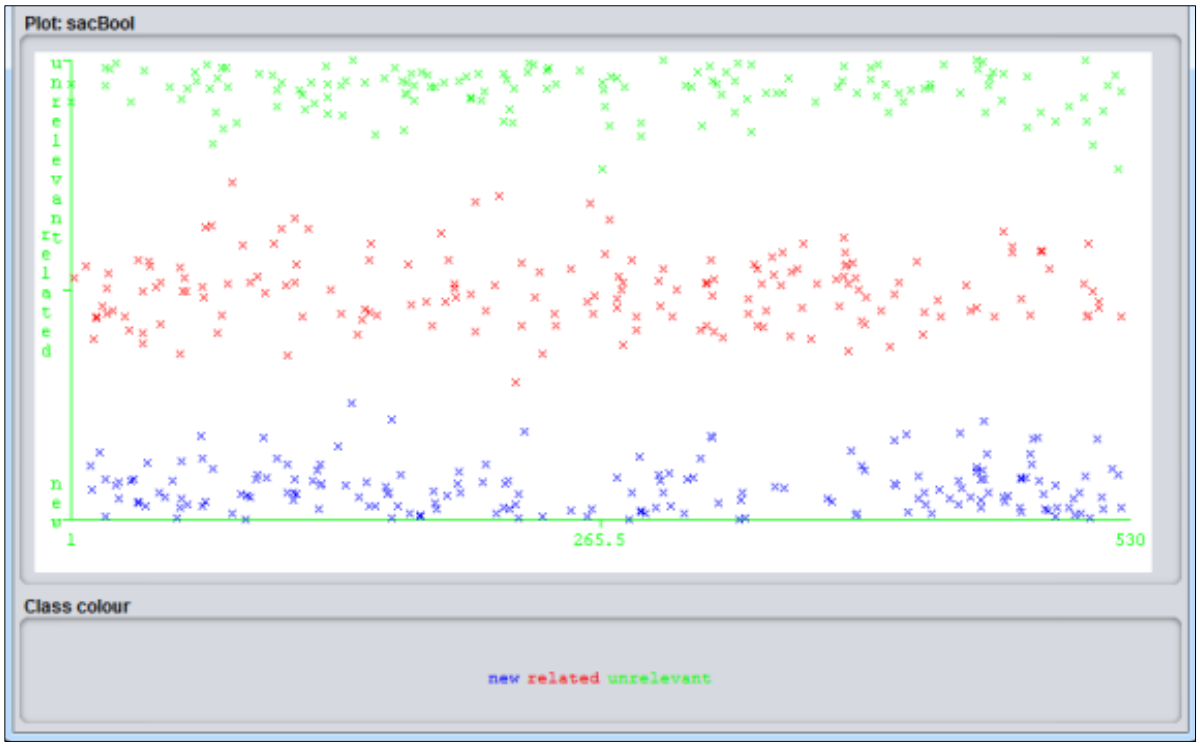

Figure 22 : visualisation résultat algorithme SMV à partir d'une matrice de valeurs booléennes

Dans la fenêtre de résultats du classifieur (figure 22), les points ou croix de la bleu, rouge ou verte représentent les documents classés respectivement comme new, related ou unrelated.

En cliquant sur l'une de ces croix, on obtient des détails concernant le point correspondant (exemple en figure 23).

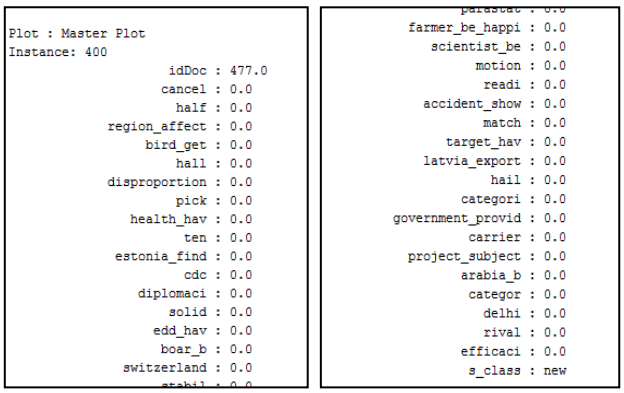

Figure 23 : visualisation instance classée

SAUVEGARDE DU MODELE

Après l'étape de test ou d'exécution d'algorithmes J48, Naïve Bayes et SVM, nous sauvegardons notre modèle en vue de son évaluation sur un autre jeu de données (les données de validations) et son intégration dans notre code ou programme java de classification automatique d'articles.

EVALUATION DE L'APPRENTISSAGE

A partir des données d'apprentissage, les résultats d'exécutions de trois algorithmes testés à savoir Naïve Bayes, SMO (SVM), et J48 (arbre de décision) sur un ensemble original d'attributs sont présentés dans le tableau 5.

	Résultat d'évaluation							
	CorrectlyClassified	IncorrectlyClassified	La macro moyenne (WeightedAvg.) sur les classes					
Macro moyenne générale			TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	58	42	0,5	0,2	0,6	0,6	0,5	0,7
SMO	64	36	0,6	0,1	0,6	0,6	0,6	0,7
J48	58	42	0,5	0,2	0,5	0,6	0,6	0,7

Tableau 5 : Moyenne générale d'évaluation de trois modèles de fonctions paramétrées. Les résultats détaillés sont présenté dans l'annexe de ce mémoire.

Au vu des résultats présentés dans le tableau ci-dessus, et dont les détails sont présentés en annexe, nous avons sélectionné SMO comme algorithme principal avec un seuil de 2 (en parlant d'élagage Df (Document frequency) sur la terminologie du corpus d'étude).

Après sélection de sous-ensemble sur l'ensemble original d'attribut, le résultat d'exécution de l'algorithme SMO sur les données est passé de 72.8 % (CorrectlyClassified Instances, avec 2291 attributs filtrés) à 82.5 % (CorrectlyClassified Instances avec 334 attributs sélectionnés). Le résultat (après traitement de doublons dans le corpus et sélection de variable) est présenté dans le tableau suivant :

		Résultat d'évaluation SMO sur le corpus d'étude							
		CorrectlyClassified	Incorrectly Classified	La moyenne (WeightedAvg.) sur les classes					
Type de filtrage	Nombre d'attributs			TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Simple	2291	73 %	27 %	0,7	0,1	0,7	0,7	0,7	0,8
WrapperBest First	334	82 %	17 %	0,8	0,1	0,8	0,8	0,8	0,9

Tableau 6 : Les résultats d'évaluation de SMO sur l'ensemble original (au nombre 2291attributs), et le sous-ensemble (au nombre de 334 attributs).

Au vu des résultats, notre approche fonctionne bien, le résultat avec le sous-ensemble est très satisfaisant par rapport à l'ensemble original d'attributs.

Intégration et restitution des résultats

Dans cette partie du manuscrit, nous présentons le processus d'implémentation de notre approche (l'algorithme d'apprentissage pour la catégorisation automatique de documents), le but est de discriminer en ligne les articles de média de la base de données du système Padiweb. Ainsi, cette partie permet d'évaluer, en production, notre classifieur (algorithme d'apprentissage machine)

1. Les principales phases de notre programme

Les deux grandes phases pour la catégorisation d'articles sont :

1.1. Entraînement du classifieur ;

1.2. Prédiction de classe des documents de la base (du Padiweb).

Dans la première phase (d'entrainement du classifieur), nous disposons d'un corpus étiqueté (pour chaque texte ou article, nous connaissons, en avance sa classe) en format json, dans notre cas. Cette collection de documents (le corpus étiqueté) est alors vectorisé (grâce au modèle de représentation textuelle, éventuellement, fondé sur la terminologie, dans notre cas), ou représenté sous la forme matricielle depuis nos codes java, et dont la sortie est un fichier en forma arff, dans notre cas.

Le fichier (en format arff) est fourni en entrée d'un algorithme d'apprentissage (sous weka, ou code java utilisant la librairie weka), qui va trouver la meilleure fonction de prédiction.

Dans la seconde phase (la phase de prédiction de classes de documents), les documents (articles de média dans notre cas) de la base de test (du Padiweb dans notre cas) sont aussi vectorisé avec le même modèle, et le classifieur appris leur assigne des étiquettes de classe, grâce au modèle de fonction de prédiction. Ainsi cette phase, permet d'évaluer ou valider le classifieur appris (Amini and Gaussier, 2013). La figure 24, illustre les deux grandes phases de notre classifieur.

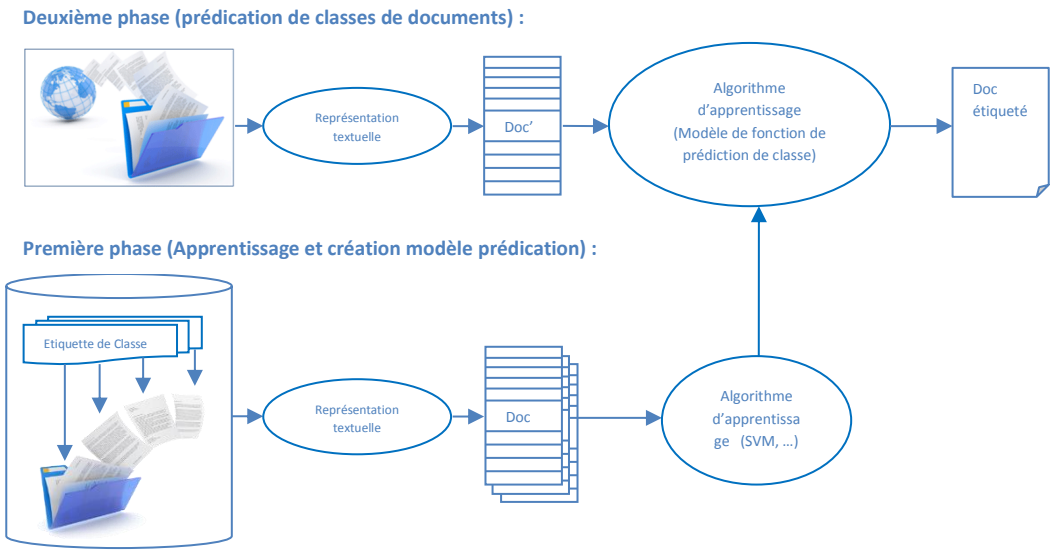

Figure 24 : les deux grandes phases de catégorisation de documents

Cirad Classifieur PadiWeb (CCP)

Le classifieur (CCP, pour Cirad Classifieur PadiWeb), développé en Java, est composé de deux grands modules (d'apprentissage ou création de modèle de fonction, et de prédiction) répartis en quatre onglets suivant :

1. **Corpus & ressources**

L'onglet Corpus & ressouces (figure 25) permet de définir respectivement l'emplacement absolu du corpus d'entrainement, ressources et Treetagger. Il permet, en outre, la définition des paramètre Treetagger.

Figure 25 : Corpus & ressources

Dans la capture écran, les champs « Corpus d'apprentissage », « Fichier de Mots controlés », « path treetagger », « paramètre treetagger » et « path wordnet », permettent de définir respectivment les chemins du corpus d'entrainement de modèle de fonction de prédiction, le fichier de mots-clés pour la représentation vectorielle avec un peu de la sémantique, l'emplacement treetagger pour notre deuxième approche d'étiquetage, les parmètres d'étiqueteur treetagger, et l'emplacement wordnet pour une éventuelle représentation conceptuelle.

2. Filtrage et Création fichier de descripteus

L'onglet Filtrage et Création de fonction de descrpteurs (figure 26), permet la selection de variables (la définition de vocabulaire) prédictives de notre espace de représentation vectorielle, et la définition de l'emplacement du fichier qui contiendra les idf de chaque descrpteurs pour l'usage ultérieure de calcul tf-idf.

Figure 26 : Filtrage & création vocabulaire

Dans la capture écran, les champs :

- « Filtrage Grammatical » permet de renseigner les patrons de filtrage.
  Par ex : Nom + Verbe, Adjectif.

- « Normalisation Surfacique par Regex» permet de renseigner les patrons de filtrage.
  Par ex : l'expression « \\n|
|[()]|NAN|\\p{Punct}|\\“|\\”|’’|\\p{Digit} » permet de normaliser de nétoyer les mots contenant de caractères qui ne sont pas alphanumérique.

- « Filtrage par Antidictionnaire » permet de renseigner l'emplacement du fichier contenant les mots vides.

- « Filtrage Statistique » permet de définir le seuil sur la mesure Df (Document frequency) : le seuillage. Cette opération (le seuillage) constiste à enlever les termes du vocabulaire présents dans l'ensemble d'entrainement (notamment dans l'espace de représentation matricielle) et qui ont une mesure Df inférieur au seuil que nous définissons dans le champs « Filtrage Statistique ».

  Par ex : 1, 2, 3, 4, 5.

3. Création de fonction de prédiction de classes d'articles

L'onglet Création de fonction de prédiction (figure 27), permet la selection de variables (la définition de vocabulaire) prédictives de notre espace de représentation vectorielle, et la définition de l'emplacement du fichier contenant les variables sélectionner par méthode wrapper sous weka :

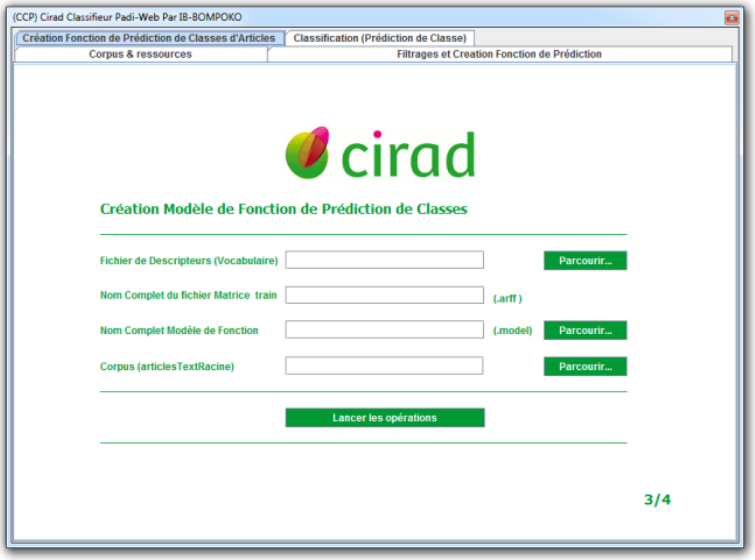

Figure 27 : Création Modèle de fonction de prédiction de classes

4. Classification automatique (Prédication des classes d'articles)

L'onglet Classification automatique (figure 28), permet la réalisation de prédictions de classed'articles de la base de données, grâce au modèle de focntion appris renseigné dans le champ « Nom complet Modèle de fonction », de fichier de descripteurs ou d'attributs avec leurs idf associés, les champs url, compte user et mot de pass, permettent l'accès à la base de données pour l'assignation de classe d'articles.

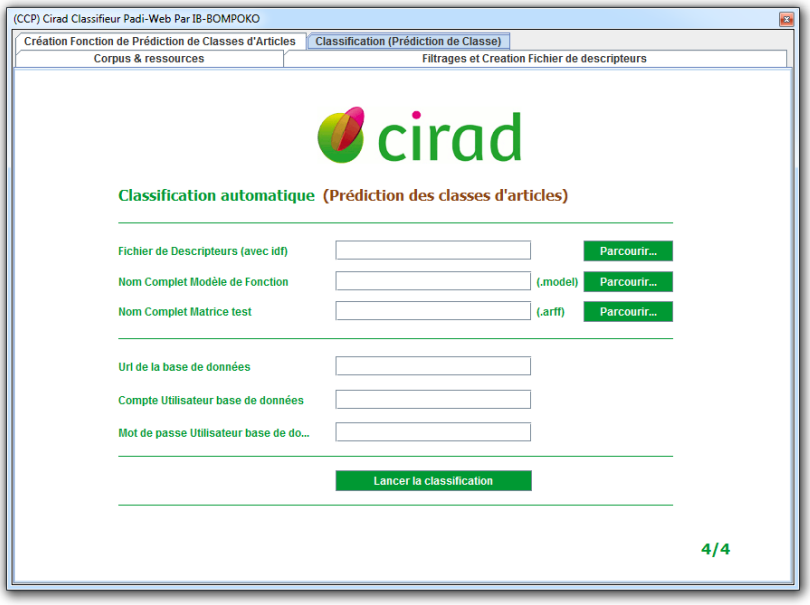

Figure 28 : Classification automatique

5. Base de données Padiweb

Afin d'insérer le resultat de procéssus de la classification automatique dans la base de données du système Padiweb, les principales opérations sont les suivantes :

- La création d'un nouveau champs denomé « classe », cf la figure 29 ;
- Ensuite, l'insertion des étiquetes des classes dans la colonne ou champ classe, cf la figure 30 .

Le resultat final est alors présenté dans la figure 31.

id	title	url	source	text	lang	probability_lang	date_submitted	date_aspirated	id_rssfeed
0003209d62	Niigata culls 540000 chickens to combat bird flu a…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	The Japan Times	Niigata culls 540,000 chickens to combat bird flu …	EN	0.32	1480825518	1480838428	075a4e9c14
00135cea13	Her dad died, and she\'s had to handle lupus and l…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Miami Herald	Wish Book: Milagros Campos is on her own with lupu…	EN	0.44	1482321669	1482393697	15125d9a6f
00154a28dd	Marco Rubio speaks out about Zika funding - NBC2 N…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	NBC2 News	Marco Rubio speaks out about Zika funding LEE…	EN	0.43	1472077188	1472109061	ad75cdc9e1
001b0970f5	Mosquito threat brings different element to outdoo…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	ournal (blog)	Mosquito threat brings different element to outdoo…	EN	0.5	1468447481	1468480361	e0a6a790e1
0021ad1558	Tips to tackle dengue - The Hindu	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	The Hindu	Tips to tackle dengue - The Hindu ith rising…	EN	0.38	1440288127	1456819439	5705f8234a

Figure 29 : Table de la base Padiweb (dépourvu du champ « classe »)

id	title	url	source	text	date_submitted	date_aspirated	id_rssfeed	class
ccb8dba997	Gene mutation causes juvenile mortality in calves …	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Phys.Org	Gene mutation causes juvenile mortality in calves\…	1467041025	1469517255	cf49dae16b	
ccbe33822c	Bird flu found in Georgia chicken flock - Atlanta …	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Atlanta Journal Constitution	Avian flu discovered in Georgia chicken flock\n\n\…	1490629732	1490943672	10c8057870	
ccc4b2c0a4	Vets warn farmers to be alert for theileria risks …	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Stuff.co.nz	Vets warn farmers to be alert for theileria risks …	1468196133	1468221049	e0a6a790e1	
cccb599234	Pembrolizumab shows promise in treatment of mesoth…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Science Daily	Contenu Introuvable.	1490029825	1491807659	15125d9a6f	
cccb93ae4d	Thanh Hoa intensifies A/H7N9 avian flu prevention …	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	SGGP	SGGP English Edition- Thanh Hoa intensifies A/H7N9…	1492059413	1492066881	10c8057870	
cccf90e55c	Evil dad who starved and tortured little boy befor…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	SDE Entertainment News	Evil dad who starved and tortured little boy befor…	1494396728	1494918391	7b34ef8790	
ccd4996950	Utilizing Dietary Cation-Anion Difference to Maxim…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Dairy Herd Management	Utilizing Dietary Cation-Anion Difference to Maxim…	1497890411	1497942844	e0a6a790e1	
ccd53beb6f	Happy Dead Duck Day - Atlas Obscura	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Atlas Obscura	Happy Dead Duck Day - Atlas Obscura\n \nKees Moelik…	1496694221	1496733081	ece86a9f0a	
ccd6281b17	A Teen\'s Family Fought To Get Her A Restricted TB…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	NPR	Goats and Soda : NPR\n\n\n \n \n …	1484854930	1484899672	7903d30a98	
ccd93b8918	Coal Ash Bedevils Oklahoma Town, Revealing Weaknes…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Daily Yonder	Coal Ash Bedevils Oklahoma Town, Revealing Weaknes…	1470393003	1470467532	cf49dae16b	
ccdab4d5be	Foyer de grippe aviaire hautement pathogène à viru…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	MesVaccins.net	Foyer de grippe aviaire hautement pathogène à viru…	1485246649	1485331442	4a64da2cb9	
cce2cbe54d	March for Life 2016 ? Scenes from the annual anti-…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Ottawa Citizen	March for Life 2016 - Scenes from the annual anti-…	1463067013	1463474861	ad75cdc9e1	

Figure 30 : Table de la base Padiweb (avec champs « classe »)

id	title	url	source	text	date_submitted	date_aspirated	id_rssfeed	class
0003209d62	Niigata culls 540000 chickens to combat bird flu a…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	The Japan Times	Niigata culls 540,000 chickens to combat bird flu …	1480825518	1480838428	075a4e9c14	unrelevant
00087e5642	Discovering Waitomo\'s glow worm wonderland - New …	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	New Zealand Herald	Discovering Waitomo\'s glow worm wonderland - Trav…	1477182626	1477206256	51fe179e08	related
0010ee6717	Corridors of Power - The Star, Kenya	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	The Star, Kenya	Corridors of Power | The Star, Kenya\n\nThe wrangl…	1471834676	1471849509	7903d30a98	unrelevant
00135cea13	Her dad died, and she\'s had to handle lupus and l…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Miami Herald	Wish Book: Milagros Campos is on her own with lupu…	1482321669	1482393697	15125d9a6f	unrelevant
00154a28dd	Marco Rubio speaks out about Zika funding - NBC2 N…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	NBC2 News	Marco Rubio speaks out about Zika funding\n\nLEE C…	1472077188	1472109061	ad75cdc9e1	unrelevant
001b0970f5	Mosquito threat brings different element to outdoo…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	ournal (blog)	Mosquito threat brings different element to outdoo…	1468447481	1468480361	e0a6a790e1	unrelevant
0021a4e2e9	Kansas Woman Gets Life Sentence In Death Of Stepso…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	The Inquisitr	Kansas Woman Gets Life Sentence In Death Of Stepso…	1479330384	1480665982	7b34ef8790	unrelevant
0021ad1558	Tips to tackle dengue - The Hindu	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	The Hindu	Tips to tackle dengue - The Hindu\n\n\n\nith risin…	1440288127	1456819439	5705f8234a	unrelevant
002287a95f	Weather warning for horse and livestock owners | V…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	vet times	Weather warning for horse and livestock owners\n\n…	1480525152	1480665777	4a0fb79fd8	unrelevant
0023458f3c	A smooth lambing season on this 370-ewe farm in Do…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Irish Independent	A smooth lambing season on this 370-ewe farm in Do…	1492837305	1492931013	6df579530d	new
0024d34de3	Grippe aviaire: oiseaux découverts dans le Chablai…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Tribune de Genève	Valais: Grippe aviaire: oiseaux infectés dans le C…	1479385815	1479456220	4a64da2cb9	unrelevant
002625fb62	Avian flu conformed in Alappuzha - Times of India	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Times of India	Avian flu conformed in Alappuzha\n\n ALAPPUZHA:…	1477393740	1477724495	10c8057870	unrelevant
00298bbd1e	Archers to the rescue in Madrid as boars trespass …	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Yahoo News	Archers to the rescue in Madrid as boars trespass\…	1483417958	1483517189	7b34ef8790	unrelevant
002aa7122f	Bluetongue levels rise in France, risk level of re…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Agriland	Bluetongue levels rise in France, risk level of re…	1479880896	1479888211	4373aa30ab	unrelevant
002c92178c	My place: Suzanne McDonnell - Star Weekly	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Star Weekly	Star Weekly | My place: Suzanne McDonnell\n\n\n …	1473091458	1473145702	7e768e403e	unrelevant
002f631440	Novus Global Swine Nutrition Roundtable forges fut…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Farm Forum	Attention Required! | Cloudflare\n\n\n …	1492540781	1492758470	a83cc6376b	unrelevant
003016c995	Government to rope in Akshaya Patra to run Hingoni…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Times of India	Government to rope in Akshaya Patra to run Hingoni…	1473213960	1473232340	c9593f6e51	new
0032a5b7f6	Watch: lameness management tips from the experts -…	http://news.google.com/news/url?sa=t&fd=R&ct2=us&u…	Irish Farmers Journal	lameness management tips from the experts 09 March…	1489012314	1489132898	09f6823154	unrelevant

Figure 31 : Table de la base Padiweb (avec champs « classe ») contenant le résultat

Conclusion

Dans ce travail, nous avons proposé et réalisé une approche de classification de documents réputés difficiles liés à l'épidémiologie animale. Nous avons alors intégré nos propositions dans un système de surveillance de propagation des maladies (plate-forme Padiweb).

Notre processus de classification (le classifieur) est construit à partir d'un corpus étiqueté en format Json, et repose sur des librairies java suivantes : «json », « tt4J, treetagger for java», « opennlp », « standford-corenlp », « snowball-stemmer », « mysql-connector » et « weka ».

La première librairie (json) permet les opérations telles que l'extraction à partir fichier d'extension en json.

Les trois librairies suivantes (« TT4J, TreeTagger for Java», «Apache opennlp », « standford-corenlp ») permettent les opérations d'étiquetages morpho-syntaxique sur les textes extraits de fichiers en format json.

La quatrième librairie « mysql-connector » permet les accès à la base de données de nouveaux articles extraits depuis les sources informelles telles que le web.

Enfin la dernière librairie permet la réalisation effective des opérations de classification.

Notre algorithme s'emploie en deux grandes étapes : l'apprentissage hors-ligne (qui se réalise une fois pour la construction de modèle de prédiction depuis le corpus d'apprentissage, dans notre cas le corpus étiqueté en json) et le classement (dite aussi de prédiction, qui exécute le modèle de prédiction issu de la phase d'apprentissage sur les nouvelles données).

Ces deux grandes étapes intègrent le processus de prétraitement de données utilisant les mêmes paramètres en apprentissage comme en prédiction.

Lors de l'étape de prétraitement, notre programme utilise une approche originale d'étiquetage de textes (nous avons écrit notre propre code java fondé sur l'étiqueteur treetagger).

Durant la phase de représentation vectorielle, notre démarche, se propose une approche originale basée sur les termes contrôlés proposés par des experts (pour notre cas, de l'unité mixe au Cirad);

Durant la phase d'élagage et de seuillage, notre programme permet les opérations dynamiques de filtrages linguistiques et statistiques sur la terminologie existante.

Lors des opérations de réduction de la taille de matrice ou vocabulaire, nous avons employé une approche de sélection de caractéristiques (ou descripteurs) fondé sur le critère d'évaluation hybride (filter et Wrapper), de stratégie de recherche (BestFirst), avec l'algorithme SVM.

Afin de classer les articles de média ou les documents dans les catégories idoines dans le système de biosurveillance, notre processus (classifieur) se décline en deux principales phases : le prétraitement des documents et la prédiction de la classe de ces documents (grâce au modèle construit à l'étape d'apprentissage à partir d'un ensemble de données étiquetées dit corpus d'apprentissage).

Nous avons testé notre classifieur sur de nouveaux articles (moissonnés au cours du stage depuis des sources informelles issues du Web). Nous avons constaté, de manière empirique, un bon comportement de notre système. Le système proposé est rapide, les règles sont simples(ou plus faciles à comprendre, et plus facile à contrôler), et les résultats sont satisfaisants.

Néanmoins il est possible d'améliorer des performances de notre classifieur, en mettant beaucoup plus l'accent sur la sémantique, les richesses de ressources lexicographiques telles que les ontologies, même si elles n'apportent pas beaucoup d'informations, dans la littérature.

Bibliographie

Kilgarriff Adam. 2006. Googleology is bad science.

Massih-Reza Amini. 2015. *Apprentissage machine: de la théorie à la pratique*. Editions Eyrolles.

Massih-Reza Amini and Éric Gaussier. 2013. *Recherche d'information: applications, modèles et algorithmes*. Editions Eyrolles.

Douglas Biber. 1993. Using Register-diversified Corpora for General Language Studies. *Comput. Linguist.*, 19(2):219–241, June.

Willem Nico Borst and W. N. Borst. 1997. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. , September.

Lynne Bowker and Jennifer Pearson. 2002. *Working with Specialized Language: A Practical Guide to Using Corpora*. Routledge. Google-Books-ID: nx4iO6boNzIC.

Antoine Cornuéjols and Laurent Miclet. 2011. *Apprentissage artificiel: Concepts et algorithmes*. Editions Eyrolles, July.

Kenneth CUKIER and Viktor MAYER-SCHOENBERGER. 2014. *Big Data: La révolution des données est en marche*. Groupe Robert laffont, February.

Gérard Dreyfus, Jean-Marc Martinez, Manuel Samuelides, Mirta B. Gordon, Fouad Badran, and Sylvie Thiria. 2011. *Apprentissage statistique: Réseaux de neurones - Cartes topologiques - Machines à vecteurs supports*. Editions Eyrolles, July.

Fabien Gandon, Olivier Corby, and Catherine Faron-Zucker. 2012. *Le web sémantique: Comment lier les données et les schémas sur le web ?*. Dunod, May.

Thomas R. Gruber. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, June.

Joel Grus. 2017. *Data science par la pratique*. Editions Eyrolles, May.

Mark F. Hornick, Erik Marcadé, and Sunil Venkayala. 2010. *Java Data Mining: Strategy, Standard, and Practice: A Practical Guide for Architecture, Design, and Implementation*. Morgan Kaufmann, July.

Pirmin Lemberger, Marc Batty, Médéric Morel, and Jean-Luc Raffaëlli. 2016a. *Big Data et Machine Learning-Manuel du data scientist-2e éd.* Dunod.

Pirmin Lemberger, Marc Batty, Médéric Morel, and Jean-Luc Raffaëlli. 2016b. *Big Data et Machine Learning-Manuel du data scientist-2e éd.* Dunod.

Michel Lutz and Eric Biernat. 2015. *Data Science : fondamentaux et études de cas: Machine Learning avec Python et R*. Editions Eyrolles, October. Google-Books-ID: 7vW4CgAAQBAJ.

Roche (Maitre de Mathieu. 2004. *Intégration de la construction de la terminologie de domaines spécialisés dans un processus global de fouille de textes*.

Patrick Naïm, Pierre-Henri Wuillemin, Philippe Leray, Olivier Pourret, and Anna Becker. 2011. *Réseaux bayésiens*. Editions Eyrolles, July. Google-Books-ID: 7d_Jq2ehb0oC.

JALAM Radwan. 2003. Apprentissage automatique et catégorisation de textes multilingues. June.

Francois Rastier. 2002. Enjeux épistémologiques de la linguistique de corpus.

John Sinclair. 1991. *Corpus, Concordance, Collocation*. Oxford University Press. Google-Books-ID: L8l4AAAAIAAJ.

TUFFERY Stéphane. 2012a. *Data Mining et statistique décisionnelle: L'intelligence des données*. Editions TECHNIP, August.

TUFFERY Stéphane. 2012b. *Data Mining et statistique décisionnelle: L'intelligence des données*. Editions TECHNIP, August. Google-Books-ID: Cs2fCgAAQBAJ.

Henry Teguiak. 2012. *Construction d'ontologies à partir de textes : une approche basée sur les transformations de modèles*. phdthesis, ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique - Poitiers, December.

Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, October.

Annexe

Evaluation et choix du modèle (entre Naïve Bayes, SMO et J48)

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms	occurrences	0	64.6825 %	35.3175 %	0,647	0,182	0,654	0,647	0,643	0,796
SMO	noms	occurrences	0	70.6349 %	29.3651 %	0,706	0,153	0,709	0,706	0,704	0,813
J48	noms	occurrences	0	66.4683 %	33.5317 %	0,665	0,172	0,664	0,665	0,663	0,752
NaïveBayes	noms	tfidf	0	71.4286 %	28.5714 %	0,714	0,149	0,732	0,714	0,716	0,806
SMO	noms	tfidf	0	74.8016 %	25.1984 %	0,748	0,130	0,755	0,748	0,749	0,848
J48	noms	tfidf	0	66.2698 %	33.7302 %	0,663	0,173	0,670	0,663	0,665	0,767
NaïveBayes	noms	booléenne	0	66.6667 %	33.3333 %	0,667	0,174	0,679	0,667	0,657	0,846
SMO	noms	booléenne	0	73.2143 %	26.7857 %	0,732	0,138	0,734	0,732	0,732	0,836
J48	noms	booléenne	0	66.2698 %	33.7302 %	0,663	0,172	0,661	0,663	0,662	0,769
NaïveBayes	noms	occurrences	1	65.2778 %	34.7222 %	0,653	0,180	0,671	0,653	0,649	0,806
SMO	noms	occurrences	1	73.0159 %	26.9841 %	0,730	0,138	0,729	0,730	0,729	0,829
J48	noms	occurrences	1	65.0794 %	34.9206 %	0,651	0,178	0,647	0,651	0,648	0,729
NaïveBayes	noms	tfidf	1	71.8254 %	28.1746 %	0,718	0,147	0,734	0,718	0,719	0,812
SMO	noms	tfidf	1	73.4127 %	26.5873 %	0,734	0,136	0,740	0,734	0,736	0,840
J48	noms	tfidf	1	67.0635 %	32.9365 %	0,671	0,169	0,676	0,671	0,673	0,764
NaïveBayes	noms	booléenne	1	66.6667 %	33.3333 %	0,667	0,174	0,679	0,667	0,657	0,846
SMO	noms	booléenne	1	72.4206 %	27.5794 %	0,724	0,142	0,726	0,724	0,725	0,831
J48	noms	booléenne	1	64.4841 %	35.5159 %	0,645	0,181	0,642	0,645	0,643	0,751

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms	occurrences	2	64.6825 %	35.3175 %	0,647	0,182	0,654	0,647	0,643	0,797
SMO	noms	occurrences	2	71.627 %	28.373 %	0,716	0,146	0,715	0,716	0,713	0,817
J48	noms	occurrences	2	65.873 %	34.127 %	0,659	0,175	0,658	0,659	0,657	0,745

NaïveBayes	noms	tfidf	2	71.4286 %	28.5714 %	0,714	0,147	0,718	0,714	0,714	0,847
SMO	noms	tfidf	2	75.3968 %	24.6032 %	0,754	0,127	0,759	0,754	0,755	0,854
J48	noms	tfidf	2	64.4841 %	35.5159 %	0,645	0,182	0,646	0,645	0,645	0,735
NaïveBayes	noms	booléenne	2	66.0714 %	33.9286 %	0,661	0,177	0,674	0,661	0,652	0,845
SMO	noms	booléenne	2	73.8095 %	26.1905 %	0,738	0,134	0,741	0,738	0,739	0,837
J48	noms	booléenne	2	64.4841 %	35.5159 %	0,645	0,181	0,642	0,645	0,643	0,751
NaïveBayes	noms	occurrences	3	64.2857 %	35.7143 %	0,643	0,185	0,650	0,643	0,638	0,794
SMO	noms	occurrences	3	72.0238 %	27.9762 %	0,720	0,144	0,720	0,720	0,718	0,818
J48	noms	occurrences	3	65.873 %	34.127 %	0,659	0,175	0,658	0,659	0,657	0,745
NaïveBayes	noms	tfidf	3	64.4841 %	35.5159 %	0,645	0,182	0,646	0,645	0,645	0,735
SMO	noms	tfidf	3	75 %	25 %	0,750	0,127	0,752	0,750	0,751	0,849
J48	noms	tfidf	3	64.4841 %	35.5159 %	0,645	0,182	0,646	0,645	0,645	0,735
NaïveBayes	noms	booléenne	3	66.0714 %	33.9286 %	0,661	0,177	0,674	0,661	0,652	0,844
SMO	noms	booléenne	3	72.8175 %	27.1825 %	0,728	0,140	0,730	0,728	0,729	0,834
J48	noms	booléenne	3	64.4841 %	35.5159 %	0,645	0,181	0,642	0,645	0,643	0,751

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms	occurrences	4	65.873 %	34.127 %	0,659	0,178	0,682	0,659	0,654	0,806
SMO	noms	occurrences	4	71.0317 %	28.9683 %	0,710	0,149	0,709	0,710	0,708	0,816
J48	noms	occurrences	4	64.6825 %	35.3175 %	0,647	0,181	0,645	0,647	0,645	0,733
NaïveBayes	noms	tfidf	4	64.881 %	35.119 %	0,649	0,182	0,656	0,649	0,644	0,798
SMO	noms	tfidf	4	71.627 %	28.373 %	0,716	0,146	0,714	0,716	0,713	0,820
J48	noms	tfidf	4	65.873 %	34.127 %	0,659	0,175	0,658	0,659	0,657	0,745
NaïveBayes	noms	booléenne	4	66.2698 %	33.7302 %	0,663	0,176	0,675	0,663	0,654	0,844
SMO	noms	booléenne	4	72.8175 %	27.1825 %	0,728	0,140	0,728	0,728	0,728	0,833
J48	noms	booléenne	4	64.6825 %	35.3175 %	0,647	0,180	0,644	0,647	0,645	0,749
NaïveBayes	noms	occurrences	5	63.8889 %	36.1111 %	0,639	0,187	0,645	0,639	0,633	0,793
SMO	noms	occurrences	5	71.2302 %	28.7698 %	0,712	0,148	0,711	0,712	0,709	0,814

J48	noms	occurrences	5	65.2778 %	34.7222 %	0,653	0,178	0,652	0,653	0,651	0,740
NaïveBayes	noms	tfidf	5	73.4127 %	26.5873 %	0,734	0,137	0,737	0,734	0,734	0,853
SMO	noms	tfidf	5	75 %	25 %	0,750	0,128	0,751	0,750	0,750	0,847
J48	noms	tfidf	5	64.4841 %	35.5159 %	0,645	0,182	0,646	0,645	0,645	0,735
NaïveBayes	noms	booléenne	5	66.4683 %	33.5317 %	0,665	0,175	0,677	0,665	0,656	0,843
SMO	noms	booléenne	5	73.2143 %	26.7857 %	0,732	0,138	0,733	0,732	0,731	0,835
J48	noms	booléenne	5	64.6825 %	35.3175 %	0,647	0,180	0,644	0,647	0,645	0,750

Evaluation et choix du modèle

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	verbes	occurrences	0	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	verbes	occurrences	0	69.0476 %	30.9524 %	0,690	0,156	0,687	0,690	0,687	0,802
J48	verbes	occurrences	0	59.3254 %	40.6746 %	0,593	0,206	0,587	0,593	0,589	0,721
NaïveBayes	verbes	tfidf	0	58.9286 %	41.0714 %	0,589	0,211	0,599	0,589	0,587	0,783
SMO	verbes	tfidf	0	68.0556 %	31.9444 %	0,681	0,163	0,679	0,681	0,679	0,795
J48	verbes	tfidf	0	63.6905 %	36.3095 %	0,637	0,184	0,632	0,637	0,633	0,726
NaïveBayes	verbes	booléenne	0	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	verbes	booléenne	0	69.0476 %	30.9524 %	0,690	0,156	0,687	0,690	0,687	0,802
J48	verbes	booléenne	0	59.3254 %	40.6746 %	0,593	0,206	0,587	0,593	0,589	0,721
NaïveBayes	verbes	occurrences	1	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	verbes	occurrences	1	68.254 %	31.746 %	0,683	0,160	0,679	0,683	0,679	0,801
J48	verbes	occurrences	1	59.5238 %	40.4762 %	0,595	0,205	0,589	0,595	0,591	0,718
NaïveBayes	verbes	tfidf	1	71.4286 %	28.5714 %	0,714	0,149	0,732	0,714	0,716	0,806
SMO	verbes	tfidf	1	74.8016 %	25.1984 %	0,748	0,130	0,755	0,748	0,749	0,848
J48	verbes	tfidf	1	66.2698 %	33.7302 %	0,663	0,173	0,670	0,663	0,665	0,767
NaïveBayes	verbes	booléenne	1	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	verbes	booléenne	1	68.254 %	31.746 %	0,683	0,160	0,679	0,683	0,679	0,801
J48	verbes	booléenne	1	59.5238 %	40.4762 %	0,595	0,205	0,589	0,595	0,591	0,718

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	verbes	occurrences	2	61.3095 %	38.6905 %	0,613	0,202	0,619	0,613	0,601	0,763
SMO	verbes	occurrences	2	70.4365 %	29.5635 %	0,704	0,150	0,700	0,704	0,700	0,808
J48	verbes	occurrences	2	60.3175 %	39.6825 %	0,603	0,200	0,599	0,603	0,600	0,740
NaïveBayes	verbes	tfidf	2	58.9286 %	41.0714 %	0,589	0,211	0,599	0,589	0,587	0,782
SMO	verbes	tfidf	2	68.4524 %	31.5476 %	0,685	0,157	0,685	0,685	0,684	0,797
J48	verbes	tfidf	2	61.9048 %	38.0952 %	0,619	0,194	0,613	0,619	0,614	0,709
NaïveBayes	verbes	booléenne	2	59.7222 %	40.2778 %	0,597	0,208	0,607	0,597	0,588	0,778
SMO	verbes	booléenne	2	67.2619 %	32.7381 %	0,673	0,164	0,669	0,673	0,669	0,795
J48	verbes	booléenne	2	59.5238 %	40.4762 %	0,595	0,205	0,589	0,595	0,591	0,718
NaïveBayes	verbes	occurrences	3	61.3095 %	38.6905 %	0,613	0,202	0,619	0,613	0,601	0,763
SMO	verbes	occurrences	3	70.0397 %	29.9603 %	0,700	0,151	0,696	0,700	0,696	0,812
J48	verbes	occurrences	3	59.7222 %	40.2778 %	0,597	0,203	0,594	0,597	0,595	0,736
NaïveBayes	verbes	tfidf	3	58.1349 %	41.8651 %	0,581	0,216	0,592	0,581	0,579	0,780
SMO	verbes	tfidf	3	67.6587 %	32.3413 %	0,677	0,164	0,675	0,677	0,675	0,791
J48	verbes	tfidf	3	62.6984 %	37.3016 %	0,627	0,189	0,622	0,627	0,623	0,719
NaïveBayes	verbes	booléenne	3	59.5238 %	40.4762 %	0,595	0,209	0,605	0,595	0,586	0,778
SMO	verbes	booléenne	3	66.8651 %	33.1349 %	0,669	0,167	0,664	0,669	0,665	0,795
J48	verbes	booléenne	3	60.5159 %	39.4841 %	0,605	0,200	0,601	0,605	0,602	0,724

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	verbes	occurrences	4	59.3254 %	40.6746 %	0,593	0,210	0,604	0,593	0,584	0,777
SMO	verbes	occurrences	4	67.0635 %	32.9365 %	0,671	0,166	0,667	0,671	0,667	0,798
J48	verbes	occurrences	4	60.5159 %	39.4841 %	0,605	0,200	0,601	0,605	0,602	0,725
NaïveBayes	verbes	tfidf	4	58.1349 %	41.8651 %	0,581	0,215	0,590	0,581	0,578	0,783
SMO	verbes	tfidf	4	66.8651 %	33.1349 %	0,669	0,168	0,664	0,669	0,665	0,789
J48	verbes	tfidf	4	62.8968 %	37.1032 %	0,629	0,188	0,624	0,629	0,626	0,721

NaïveBayes	verbes	booléenne	4	59.3254 %	40.6746 %	0,593	0,210	0,604	0,593	0,584	0,777
SMO	verbes	booléenne	4	67.0635 %	32.9365 %	0,671	0,166	0,667	0,671	0,667	0,798
J48	verbes	booléenne	4	60.5159 %	39.4841 %	0,605	0,200	0,601	0,605	0,602	0,725
NaïveBayes	verbes	occurrences	5	60.9127 %	39.0873 %	0,609	0,204	0,615	0,609	0,596	0,761
SMO	verbes	occurrences	5	71.0317 %	28.9683 %	0,710	0,144	0,706	0,710	0,707	0,814
J48	verbes	occurrences	5	58.3333 %	41.6667 %	0,583	0,210	0,580	0,583	0,581	0,722
NaïveBayes	verbes	tfidf	5	58.9286 %	41.0714 %	0,589	0,211	0,598	0,589	0,586	0,790
SMO	verbes	tfidf	5	67.2619 %	32.7381 %	0,673	0,164	0,670	0,673	0,671	0,787
J48	verbes	tfidf	5	63.2937 %	36.7063 %	0,633	0,186	0,628	0,633	0,630	0,727
NaïveBayes	verbes	booléenne	5	59.127 %	40.873 %	0,591	0,211	0,602	0,591	0,582	0,777
SMO	verbes	booléenne	5	66.6667 %	33.3333 %	0,667	0,167	0,663	0,667	0,663	0,793
J48	verbes	booléenne	5	59.127 %	40.873 %	0,591	0,211	0,602	0,591	0,582	0,777

Evaluation et choix du modèle (à corriger)

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms + verbes	occurrences	0	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	noms + verbes	occurrences	0	69.0476 %	30.9524 %	0,690	0,156	0,687	0,690	0,687	0,802
J48	noms + verbes	occurrences	0	59.3254 %	40.6746 %	0,593	0,206	0,587	0,593	0,589	0,721
NaïveBayes	noms + verbes	tfidf	0	55.1587 %	44.8413 %	0,552	0,217	0,562	0,552	0,543	0,719
SMO	noms + verbes	tfidf	0	57.1429 %	42.8571 %	0,571	0,219	0,565	0,571	0,561	0,695
J48	noms + verbes	tfidf	0	53.1746 %	46.8254 %	0,532	0,223	0,555	0,532	0,527	0,678
NaïveBayes	noms + verbes	booléenne	0	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	noms + verbes	booléenne	0	69.0476 %	30.9524 %	0,690	0,156	0,687	0,690	0,687	0,802
J48	noms + verbes	booléenne	0	59.3254 %	40.6746 %	0,593	0,206	0,587	0,593	0,589	0,721
NaïveBayes	noms + verbes	occurrences	1	61.3095 %	38.6905 %	0,613	0,202	0,619	0,613	0,601	0,763
SMO	noms + verbes	occurrences	1	70.2381 %	29.7619 %	0,702	0,151	0,697	0,702	0,696	0,807
J48	noms + verbes	occurrences	1	60.119 %	39.881 %	0,601	0,201	0,597	0,601	0,598	0,740
NaïveBaye	noms +	tfidf	1	55.1587	44.8413 %	0,55	0,21	0,562	0,552	0,543	0,71

s	verbes			%		2	7				9
SMO	noms + verbes	tfidf	1	56.3492 %	43.6508 %	0,563	0,219	0,561	0,563	0,562	0,704
J48	noms + verbes	tfidf	1	52.5794 %	47.4206 %	0,526	0,227	0,548	0,526	0,523	0,679
NaïveBayes	noms + verbes	booléenne	1	60.119 %	39.881 %	0,601	0,206	0,610	0,601	0,592	0,778
SMO	noms + verbes	booléenne	1	68.254 %	31.746 %	0,683	0,160	0,679	0,683	0,679	0,801
J48	noms + verbes	booléenne	1	59.5238 %	40.4762 %	0,595	0,205	0,589	0,595	0,591	0,718

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms + verbes	occurrences	2	61.3095 %	38.6905 %	0,613	0,202	0,619	0,613	0,601	0,763
SMO	noms + verbes	occurrences	2	70.4365 %	29.5635 %	0,704	0,150	0,700	0,704	0,700	0,808
J48	noms + verbes	occurrences	2	60.3175 %	39.6825 %	0,603	0,200	0,599	0,603	0,600	0,740
NaïveBayes	noms + verbes	tfidf	2	61.9048 %	38.0952 %	0,619	0,190	0,615	0,619	0,616	0,767
SMO	noms + verbes	tfidf	2	61.1111 %	38.8889 %	0,611	0,192	0,614	0,611	0,611	0,744
J48	noms + verbes	tfidf	2	57.7381 %	42.2619 %	0,577	0,206	0,580	0,577	0,567	0,696
NaïveBayes	noms + verbes	booléenne	2	52.381 %	47.619 %	0,524	0,252	0,636	0,524	0,476	0,758
SMO	noms + verbes	booléenne	2	63.2937 %	36.7063 %	0,633	0,182	0,633	0,633	0,632	0,751
J48	noms + verbes	booléenne	2	56.3492 %	43.6508 %	0,563	0,214	0,561	0,563	0,552	0,676
NaïveBayes	noms + verbes	occurrences	3	54.1667 %	45.8333 %	0,542	0,241	0,582	0,542	0,515	0,727
SMO	noms + verbes	occurrences	3	64.0873 %	35.9127 %	0,641	0,179	0,636	0,641	0,635	0,756
J48	noms + verbes	occurrences	3	58.5317 %	41.4683 %	0,585	0,204	0,585	0,585	0,576	0,713
NaïveBayes	noms + verbes	tfidf	3	54.1667 %	45.8333 %	0,542	0,241	0,582	0,542	0,515	0,727
SMO	noms + verbes	tfidf	3	64.0873 %	35.9127 %	0,641	0,179	0,636	0,641	0,635	0,756
J48	noms + verbes	tfidf	3	58.5317 %	41.4683 %	0,585	0,204	0,585	0,585	0,576	0,713
NaïveBayes	noms + verbes	booléenne	3	51.1905 %	48.8095 %	0,512	0,258	0,625	0,512	0,464	0,753
SMO	noms + verbes	booléenne	3	61.9048 %	38.0952 %	0,619	0,190	0,615	0,619	0,615	0,748
J48	noms + verbes	booléenne	3	58.1349 %	41.8651 %	0,581	0,205	0,580	0,581	0,570	0,693

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms + verbes	occurrences	4	53.373 %	46.627 %	0,534	0,246	0,578	0,534	0,508	0,725
SMO	noms + verbes	occurrences	4	60.119 %	39.881 %	0,601	0,199	0,595	0,601	0,595	0,730
J48	noms + verbes	occurrences	4	56.9444 %	43.0556 %	0,569	0,212	0,568	0,569	0,562	0,701
NaïveBayes	noms + verbes	tfidf	4	60.3175 %	39.6825 %	0,603	0,198	0,597	0,603	0,597	0,757
SMO	noms + verbes	tfidf	4	58.5317 %	41.4683 %	0,585	0,206	0,588	0,585	0,585	0,723
J48	noms + verbes	tfidf	4	59.3254 %	40.6746 %	0,593	0,201	0,592	0,593	0,586	0,713
NaïveBayes	noms + verbes	booléenne	4	50.7937 %	49.2063 %	0,508	0,260	0,620	0,508	0,458	0,748
SMO	noms + verbes	booléenne	4	62.6984 %	37.3016 %	0,627	0,186	0,625	0,627	0,624	0,753
J48	noms + verbes	booléenne	4	59.9206 %	40.0794 %	0,599	0,196	0,600	0,599	0,590	0,707
NaïveBayes	noms + verbes	occurrences	5	49.8016 %	50.1984 %	0,498	0,266	0,610	0,498	0,449	0,744
SMO	noms + verbes	occurrences	5	61.5079 %	38.4921 %	0,615	0,190	0,616	0,615	0,612	0,733
J48	noms + verbes	occurrences	5	58.7302 %	41.2698 %	0,587	0,202	0,586	0,587	0,578	0,702
NaïveBayes	noms + verbes	tfidf	5	59.7222 %	40.2778 %	0,597	0,200	0,593	0,597	0,591	0,758
SMO	noms + verbes	tfidf	5	58.1349 %	41.8651 %	0,581	0,207	0,584	0,581	0,581	0,717
J48	noms + verbes	tfidf	5	58.9286 %	41.0714 %	0,589	0,202	0,588	0,589	0,580	0,705
NaïveBayes	noms + verbes	booléenne	5	49.8016 %	50.1984 %	0,498	0,266	0,610	0,498	0,449	0,744
SMO	noms + verbes	booléenne	5	61.5079 %	38.4921 %	0,615	0,190	0,616	0,615	0,612	0,733
J48	noms + verbes	booléenne	5	58.7302 %	41.2698 %	0,587	0,202	0,586	0,587	0,578	0,702

Evaluation et choix du modèle

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms + verbes +adjectifs	occurrences	0	39.2857 %	60.7143 %	0,393	0,334	0,534	0,393	0,311	0,589
SMO	noms + verbes +adjectifs	occurrences	0	45.8333 %	54.1667 %	0,458	0,299	0,662	0,458	0,384	0,587
J48	noms + verbes +adjectifs	occurrences	0	39.2857 %	60.7143 %	0,393	0,327	0,420	0,393	0,351	0,520
NaïveBayes	noms + verbes	tfidf	0	40.6746 %	59.3254 %	0,407	0,310	0,610	0,407	0,318	0,599

	+adjectifs										
SMO	noms + verbes +adjectifs	tfidf	0	47.8175 %	52.1825 %	0,478	0,284	0,551	0,478	0,427	0,605
J48	noms + verbes +adjectifs	tfidf	0	39.4841 %	60.5159 %	0,395	0,324	0,410	0,395	0,359	0,528
NaïveBayes	noms + verbes +adjectifs	booléenne	0	42.8571 %	57.1429 %	0,429	0,313	0,589	0,429	0,365	0,612
SMO	noms + verbes +adjectifs	booléenne	0	46.627 %	53.373 %	0,466	0,286	0,500	0,466	0,421	0,608
J48	noms + verbes +adjectifs	booléenne	0	40.873 %	59.127 %	0,409	0,318	0,437	0,409	0,374	0,545
NaïveBayes	noms + verbes +adjectifs	occurrences	1	39.4841 %	60.5159 %	0,395	0,333	0,536	0,395	0,314	0,589
SMO	noms + verbes +adjectifs	occurrences	1	46.8254 %	53.1746 %	0,468	0,277	0,673	0,468	0,419	0,609
J48	noms + verbes +adjectifs	occurrences	1	39.4841 %	60.5159 %	0,395	0,326	0,424	0,395	0,354	0,523
NaïveBayes	noms + verbes +adjectifs	tfidf	1	40.6746 %	59.3254 %	0,407	0,310	0,610	0,407	0,318	0,599
SMO	noms + verbes +adjectifs	tfidf	1	44.246 %	55.754 %	0,442	0,299	0,530	0,442	0,422	0,589
J48	noms + verbes +adjectifs	tfidf	1	39.4841 %	60.5159 %	0,395	0,324	0,410	0,395	0,359	0,528
NaïveBayes	noms + verbes +adjectifs	booléenne	1	42.0635 %	57.9365 %	0,421	0,317	0,582	0,421	0,360	0,612
SMO	noms + verbes +adjectifs	booléenne	1	44.8413 %	55.1587 %	0,448	0,294	0,463	0,448	0,415	0,594
J48	noms + verbes +adjectifs	booléenne	1	40.873 %	59.127 %	0,409	0,318	0,437	0,409	0,374	0,545

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms + verbes +adjectifs	occurrences	2	38.4921 %	61.5079 %	0,385	0,339	0,523	0,385	0,302	0,561
SMO	noms + verbes +adjectifs	occurrences	2	40.6746 %	59.3254 %	0,407	0,312	0,551	0,407	0,345	0,556
J48	noms + verbes +adjectifs	occurrences	2	39.4841 %	60.5159 %	0,395	0,326	0,419	0,395	0,353	0,520
NaïveBayes	noms + verbes	tfidf	2	39.6825 %	60.3175 %	0,397	0,316	0,593	0,397	0,302	0,571

	+adjectifs										
SMO	noms + verbes +adjectifs	tfidf	2	40.0794 %	59.9206 %	0,401	0,324	0,547	0,401	0,370	0,551
J48	noms + verbes +adjectifs	tfidf	2	39.881 %	60.119 %	0,399	0,323	0,419	0,399	0,362	0,534
NaïveBayes	noms + verbes +adjectifs	booléenne	2	40.6746 %	59.3254 %	0,407	0,325	0,569	0,407	0,340	0,581
SMO	noms + verbes +adjectifs	booléenne	2	41.6667 %	58.3333 %	0,417	0,317	0,457	0,417	0,359	0,565
J48	noms + verbes +adjectifs	booléenne	2	41.0714 %	58.9286 %	0,411	0,317	0,436	0,411	0,376	0,553
NaïveBayes	noms + verbes +adjectifs	occurrences	3	38.4921 %	61.5079 %	0,385	0,339	0,523	0,385	0,302	0,556
SMO	noms + verbes +adjectifs	occurrences	3	39.4841 %	60.5159 %	0,395	0,320	0,531	0,395	0,347	0,545
J48	noms + verbes +adjectifs	occurrences	3	39.2857 %	60.7143 %	0,393	0,327	0,419	0,393	0,350	0,517
NaïveBayes	noms + verbes +adjectifs	tfidf	3	39.881 %	60.119 %	0,399	0,315	0,603	0,399	0,302	0,570
SMO	noms + verbes +adjectifs	tfidf	3	39.4841 %	60.5159 %	0,395	0,330	0,522	0,395	0,349	0,544
J48	noms + verbes +adjectifs	tfidf	3	39.881 %	60.119 %	0,399	0,323	0,421	0,399	0,361	0,538
NaïveBayes	noms + verbes +adjectifs	booléenne	3	40.6746 %	59.3254 %	0,407	0,325	0,568	0,407	0,339	0,576
SMO	noms + verbes +adjectifs	booléenne	3	41.4683 %	58.5317 %	0,415	0,319	0,469	0,415	0,354	0,564
J48	noms + verbes +adjectifs	booléenne	3	40.6746 %	59.3254 %	0,407	0,319	0,433	0,407	0,370	0,546

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	noms + verbes +adjectifs	occurrences	4	38.2937 %	61.7063 %	0,383	0,340	0,519	0,383	0,299	0,554
SMO	noms + verbes +adjectifs	occurrences	4	38.6905 %	61.3095 %	0,387	0,326	0,519	0,387	0,355	0,538
J48	noms + verbes +adjectifs	occurrences	4	39.6825 %	60.3175 %	0,397	0,325	0,428	0,397	0,355	0,523
NaïveBayes	noms + verbes	tfidf	4	40.2778 %	59.7222 %	0,403	0,312	0,574	0,403	0,320	0,571

	+adjectifs										
SMO	noms + verbes +adjectifs	tfidf	4	37.6984 %	62.3016 %	0,377	0,340	0,476	0,377	0,316	0,536
J48	noms + verbes +adjectifs	tfidf	4	39.6825 %	60.3175 %	0,397	0,324	0,421	0,397	0,359	0,534
NaïveBayes	noms + verbes +adjectifs	booléenne	4	40.6746 %	59.3254 %	0,407	0,325	0,568	0,407	0,339	0,574
SMO	noms + verbes +adjectifs	booléenne	4	41.6667 %	58.3333 %	0,417	0,318	0,477	0,417	0,357	0,567
J48	noms + verbes +adjectifs	booléenne	4	41.4683 %	58.5317 %	0,415	0,315	0,451	0,415	0,378	0,556
NaïveBayes	noms + verbes +adjectifs	occurrences	5	38.2937 %	61.7063 %	0,383	0,340	0,519	0,383	0,299	0,552
SMO	noms + verbes +adjectifs	occurrences	5	38.6905 %	61.3095 %	0,387	0,331	0,513	0,387	0,360	0,535
J48	noms + verbes +adjectifs	occurrences	5	39.6825 %	60.3175 %	0,397	0,325	0,421	0,397	0,355	0,527
NaïveBayes	noms + verbes +adjectifs	tfidf	5	39.881 %	60.119 %	0,399	0,315	0,565	0,399	0,314	0,569
SMO	noms + verbes +adjectifs	tfidf	5	37.8968 %	62.1032 %	0,379	0,339	0,480	0,379	0,320	0,531
J48	noms + verbes +adjectifs	tfidf	5	39.6825 %	60.3175 %	0,397	0,324	0,415	0,397	0,359	0,538
NaïveBayes	noms + verbes +adjectifs	booléenne	5	41.2698 %	58.7302 %	0,413	0,322	0,574	0,413	0,343	0,573
SMO	noms + verbes +adjectifs	booléenne	5	41.6667 %	58.3333 %	0,417	0,318	0,477	0,417	0,357	0,566
J48	noms + verbes +adjectifs	booléenne	5	41.4683 %	58.5317 %	0,415	0,315	0,444	0,415	0,378	0,560

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	N, V, A, N+V, N + V+A	occurrences	0	65.4762 %	34.5238 %	0,655	0,180	0,675	0,655	0,651	0,805
SMO	N, V, A, N+V, N + V+A	occurrences	0	73.2143 %	26.7857 %	0,732	0,137	0,731	0,732	0,731	0,826
J48	N, V, A, N+V, N + V+A	occurrences	0	64.4841 %	35.5159 %	0,645	0,182	0,646	0,645	0,645	0,738
NaïveBayes	N, V, A, N+V, N + V+A	tfidf	0	71.0317 %	28.9683 %	0,710	0,151	0,730	0,710	0,711	0,805
SMO	N, V, A,	tfidf	0	75 %	25 %	0,75	0,12	0,756	0,750	0,751	0,84

	N+V, N + V+A					0	9				8
J48	N, V, A, N+V, N + V+A	tfidf	0	61.9048 %	38.0952 %	0,619	0,195	0,622	0,619	0,621	0,727
NaïveBayes	N, V, A, N+V, N + V+A	booléenne	0	65.873 %	34.127 %	0,659	0,180	0,695	0,659	0,643	0,835
SMO	N, V, A, N+V, N + V+A	booléenne	0	74.8016 %	25.1984 %	0,748	0,130	0,752	0,748	0,749	0,848
J48	N, V, A, N+V, N + V+A	booléenne	0	64.6825 %	35.3175 %	0,647	0,181	0,650	0,647	0,648	0,730
NaïveBayes	N, V, A, N+V, N + V+A	occurrences	1	65.2778 %	34.7222 %	0,653	0,180	0,672	0,653	0,649	0,808
SMO	N, V, A, N+V, N + V+A	occurrences	1	72.8175 %	27.1825 %	0,728	0,139	0,727	0,728	0,727	0,828
J48	N, V, A, N+V, N + V+A	occurrences	1	65.2778 %	34.7222 %	0,653	0,177	0,649	0,653	0,650	0,733
NaïveBayes	N, V, A, N+V, N + V+A	tfidf	1	70.4365 %	29.5635 %	0,704	0,154	0,720	0,704	0,705	0,810
SMO	N, V, A, N+V, N + V+A	tfidf	1	73.0159 %	26.9841 %	0,730	0,138	0,736	0,730	0,732	0,839
J48	N, V, A, N+V, N + V+A	tfidf	1	61.9048 %	38.0952 %	0,619	0,195	0,624	0,619	0,621	0,725
NaïveBayes	N, V, A, N+V, N + V+A	booléenne	1	65.873 %	34.127 %	0,659	0,180	0,695	0,659	0,643	0,835
SMO	N, V, A, N+V, N + V+A	booléenne	1	74.8016 %	25.1984 %	0,748	0,130	0,752	0,748	0,749	0,848
J48	N, V, A, N+V, N + V+A	booléenne	1	64.6825 %	35.3175 %	0,647	0,181	0,650	0,647	0,648	0,730

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	N, V, A, N+V, N + V+A	occurrences	2	65.2778 %	34.7222 %	0,653	0,180	0,672	0,653	0,649	0,809
SMO	N, V, A, N+V, N + V+A	occurrences	2	72.4206 %	27.5794 %	0,724	0,142	0,723	0,724	0,722	0,828
J48	N, V, A, N+V, N + V+A	occurrences	2	65.2778 %	34.7222 %	0,653	0,177	0,649	0,653	0,650	0,733
NaïveBayes	N, V, A, N+V, N + V+A	tfidf	2	70.2381 %	29.7619 %	0,702	0,155	0,718	0,702	0,703	0,808
SMO	N, V, A,	tfidf	2	73.4127	26.5873 %	0,73	0,13	0,739	0,734	0,735	0,83

	N+V, N + V+A			%		4	7				8
J48	N, V, A, N+V, N + V+A	tfidf	2	61.9048 %	38.0952 %	0,619	0,195	0,624	0,619	0,621	0,725
NaïveBayes	N, V, A, N+V, N + V+A	booléenne	2	65.873 %	34.127 %	0,659	0,180	0,695	0,659	0,643	0,834
SMO	N, V, A, N+V, N + V+A	booléenne	2	74.6032 %	25.3968 %	0,746	0,131	0,748	0,746	0,747	0,845
J48	N, V, A, N+V, N + V+A	booléenne	2	65.2778 %	34.7222 %	0,653	0,178	0,655	0,653	0,653	0,744
NaïveBayes	N, V, A, N+V, N + V+A	occurrences	3	64.881 %	35.119 %	0,649	0,182	0,666	0,649	0,644	0,806
SMO	N, V, A, N+V, N + V+A	occurrences	3	72.2222 %	27.7778 %	0,722	0,142	0,720	0,722	0,719	0,824
J48	N, V, A, N+V, N + V+A	occurrences	3	65.2778 %	34.7222 %	0,653	0,177	0,649	0,653	0,650	0,733
NaïveBayes	N, V, A, N+V, N + V+A	tfidf	3	69.6429 %	30.3571 %	0,696	0,158	0,715	0,696	0,697	0,806
SMO	N, V, A, N+V, N + V+A	tfidf	3	74.2063 %	25.7937 %	0,742	0,133	0,748	0,742	0,744	0,843
J48	N, V, A, N+V, N + V+A	tfidf	3	61.9048 %	38.0952 %	0,619	0,195	0,624	0,619	0,621	0,725
NaïveBayes	N, V, A, N+V, N + V+A	booléenne	3	65.873 %	34.127 %	0,659	0,180	0,695	0,659	0,643	0,833
SMO	N, V, A, N+V, N + V+A	booléenne	3	75.7937 %	24.2063 %	0,758	0,125	0,758	0,758	0,758	0,850
J48	N, V, A, N+V, N + V+A	booléenne	3	65.2778 %	34.7222 %	0,653	0,178	0,655	0,653	0,653	0,744

Algorithme	Règle grammaire	Matrice de poids	Seuillage (au moins)	Résultat							
				Correctly Classified	Incorrectly Classified	La moyenne (Weighted Avg.) sur les classes					
						TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NaïveBayes	N, V, A, N+V, N + V+A	occurrences	4	65.2778 %	34.7222 %	0,653	0,181	0,675	0,653	0,647	0,809
SMO	N, V, A, N+V, N + V+A	occurrences	4	71.4286 %	28.5714 %	0,714	0,147	0,713	0,714	0,711	0,821
J48	N, V, A, N+V, N + V+A	occurrences	4	64.6825 %	35.3175 %	0,647	0,181	0,645	0,647	0,645	0,735
NaïveBayes	N, V, A, N+V, N + V+A	tfidf	4	71.2302 %	28.7698 %	0,712	0,151	0,735	0,712	0,712	0,821
SMO	N, V, A, N+V, N + V+A	tfidf	4	74.2063 %	25.7937 %	0,742	0,133	0,750	0,742	0,744	0,848

J48	N, V, A, N+V, N + V+A	tfidf	4	63.2937 %	36.7063 %	0,633	0,187	0,639	0,633	0,635	0,732
NaïveBayes	N, V, A, N+V, N + V+A	booléenne	4	65.873 %	34.127 %	0,659	0,180	0,695	0,659	0,643	0,832
SMO	N, V, A, N+V, N + V+A	booléenne	4	75.1984 %	24.8016 %	0,752	0,128	0,753	0,752	0,752	0,849
J48	N, V, A, N+V, N + V+A	booléenne	4	65.2778 %	34.7222 %	0,653	0,178	0,655	0,653	0,653	0,755
NaïveBayes	N, V, A, N+V, N + V+A	occurrences	5	65.0794 %	34.9206 %	0,651	0,182	0,674	0,651	0,645	0,810
SMO	N, V, A, N+V, N + V+A	occurrences	5	70.4365 %	29.5635 %	0,704	0,152	0,701	0,704	0,700	0,819
J48	N, V, A, N+V, N + V+A	occurrences	5	64.6825 %	35.3175 %	0,647	0,181	0,645	0,647	0,645	0,735
NaïveBayes	N, V, A, N+V, N + V+A	tfidf	5	71.627 %	28.373 %	0,716	0,149	0,741	0,716	0,717	0,822
SMO	N, V, A, N+V, N + V+A	tfidf	5	74.0079 %	25.9921 %	0,740	0,134	0,743	0,740	0,741	0,845
J48	N, V, A, N+V, N + V+A	tfidf	5	63.2937 %	36.7063 %	0,633	0,187	0,639	0,633	0,635	0,732
NaïveBayes	N, V, A, N+V, N + V+A	booléenne	5	66.0714 %	33.9286 %	0,661	0,179	0,696	0,661	0,645	0,831
SMO	N, V, A, N+V, N + V+A	booléenne	5	75.1984 %	24.8016 %	0,752	0,128	0,753	0,752	0,752	0,848
J48	N, V, A, N+V, N + V+A	booléenne	5	65.6746 %	34.3254 %	0,657	0,176	0,660	0,657	0,658	0,758

MOYENNE GENERALE

	Résultat							
	Correctly Classified	**Incorrectly Classified**	**La moyenne (Weighted Avg.) sur les classes**					
Moyenne générale			**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**
NaïveBayes	**58,1481489**	**41,8518511**	**0,58152222**	**0,21966667**	**0,63138889**	**0,58152222**	**0,55735556**	**0,74837778**
SMO	**64,0762811**	**35,9237189**	**0,64071111**	**0,18595556**	**0,66092222**	**0,64071111**	**0,62953333**	**0,75827778**
J48	**57,6258061**	**42,3741939**	**0,57629469**	**0,21806296**	**0,58142654**	**0,57629469**	**0,56601506**	**0,69031531**