REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

DEPARTEMENT Mathématiques et Informatique

N° d'ordre : ……

FILIERE : Informatique

OPTION : INTELLEGENCE ARTIFICIELLE

Mémoire présenté pour l'obtention

Du diplôme de Master

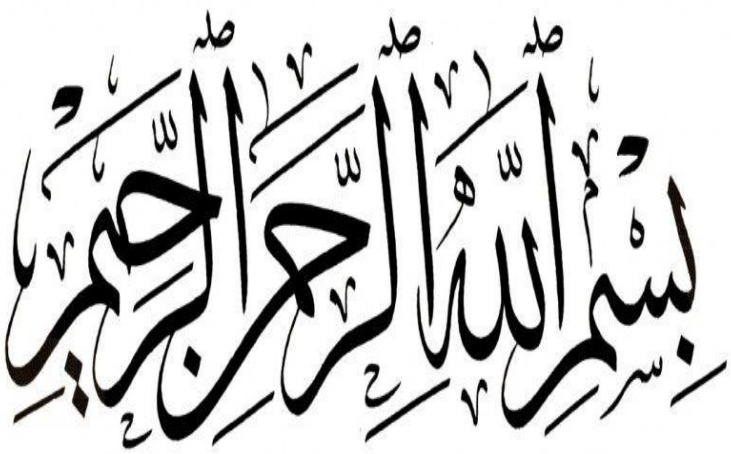
Par:Abd Eldjebar Charif & Abd Elnasser Chenene

Fouille de textes appliquée au Saint Coran

Soutenu devant le jury composé de:

Dr. KAMEL mohamed	*Université de M'sila*	*Président*
Dr. BRAHIMI Belgacem	*Université de M'sila*	*Rapporteur*
Dr. Boughraraseddik	*Université de M'sila*	*Examinateur*

Année universitaire : 2021 / 2022


بسم الله الرحمن الرحيم

Dédicaces

Je dédie ce travail :

- *A nos cher père et ma chère mère.*
- *A nos chers frères et sœurs.*
- *A toute nos famille.*
- *A toute nos amis*

A tous ceux qui ont sacrifié leur temps pour la science et à tous ceux qui utilisent la science pour le bien et la prospérité de l'humanité

Remerciements

Tout d'abord, nous remercions Dieu Tout-Puissant pour la sagesse et la connaissance sans fin.Nous tenons à remercier notre encadrant, M. Brahmi Belgacem, pour le grand honneur qu'il nousa fait c'est en nous fournissant le sujet de cette thèse. Nous avons l'honneur et le privilège travaillez avec son aide et bénéficiez de son humanité, de son professionnalisme et grâce à sa vaste expérience, il nous a guidé tout au long du processus de travail. et évolution coexistence et énergie sont ses mots. J'espère qu'il sera satisfait de cet humble acte références, pour exprimer notre gratitude et notre appréciation pour l'aide et l'appréciation les conseils qu'il nous a prodigués et les connaissances qu'il nous a transmises.

Nous sommes reconnaissants à tous nos professeurs à l'Université de M'Sila.

Nous tenons également à remercier les membres du jury d'avoir accepté le verdict de mon pays un travail.

Nous tenons à remercier toute notre famille, en particulier nos parents, qui ont toujours été nous avons été accompagnés pendant nos études. Puissent-ils trouver ici le fruit de la patience et du soutien ils ont traversé tous les moments difficiles.

Table des matières

Table des Figures

Liste des Tableaux

Liste des abréviations

TAL : Traitement automatique des langues.

RI : Recherche d'information.

EI : L'extraction de l'information.

CAH : Classification Ascendante Hiérarchique.

CT : Classification auTomatique.

RD : Recherche documentaire.

TF : Term Frequency.

IDF : Inverse Document Frequency.

TF*IDF : Term Frequency Inverse Document Frequency.

SVM : Machines à support de vecteurs.

RNA : Réseaux de neurone artificiel.

NB : Naïve Bayes.

KNN : K-nearest neighbors.

Introduction générale

Aujourd'hui, en raison de la nécessité d'une gestion automatique (classification, extraction…) des documents disponibles sur le web, la quantité d'informations obtenues par voie électronique ne cesse d'augmenter, concevoir et mettre en œuvre des outils efficaces, notamment pour permettre aux utilisateurs il devient alors absolument nécessaire de n'avoir accès qu'aux informations qu'il juge pertinentes.

Malheureusement, les travaux scientifiques qui portent sur la fouille de textes à savoir la classification, appliqués sur les textes coraniques sont très rares. La classification est parmi les tâches les plus importances dans le domaine de la fouille de textes. Car elle peut servir pour d'autres techniques telles que le filtrage de données et la recherche d'information. La classification de textes consiste à affecter un document textuel à une classe donnée selon con contenu ou autres caractéristiques.

Ce travail vise à comprendre les apports des techniques de fouille de textes telles que le Prétraitement et la Classification supervisée dans les textes du saint Coran. En particulier, nous réalisons un système automatique qui permet la classification des versets coraniques (الآيات القرانية) sur la base se leurs contenus textuels.

Objectifs de l'étude :

- Concevoir une application de classification des versets coraniques.
- Montrer le rôle de la phase de pré-traitement et représentation de textes dans la qualité de classification.
- Mener une étude comparative sur l'ensemble des algorithmes de classification afin de déterminer le meilleur algorithme pour la catégorisation des textes coraniques.

Chapitre 01 : Fouille de texte (Texte Mining)

1.1.Introduction

Le texte représente une grande quantité de différents types d'informations, et la façon dont ces informations sont présentées rend l'analyse automatique difficile. Par conséquent, l'information est non structurée (texte libre). L'absence de cette structure ne permet pas un accès direct à l'information. La quantité de données est si grande que l'analyse humaine n'est pas possible.

1.2.Définition de la fouille de texte :

L'exploration de texte, également connue sous le nom d'exploration de texte ou d'extraction de texte, est un ensemble de méthodes, de techniques et d'outils permettant d'exploiter des documents non structurés écrits sous forme de texte, tels que : L'exploration de texte s'appuie sur des techniques d'analyse linguistique pour extraire le sens de documents on structurés. Le texte Manning est utilisé pour classer des documents, créer des résumés exécutifs automatisés ou soutenir une veille stratégique ou technique le long d'un chemin de recherche prédéfini [1].

Il peut être grossièrement énoncé comme suit

TEXT MINING = LINGUISTIQUE + DATA MINING

1.3.Les tâches de Text Mining :

L'exploration de texte ne remplace pas la recherche d'informations ou le traitement du langage naturel. Les techniques qui permettent d'organiser un corpus de documents texte en fonction de leur contenu ont un très large éventail d'utilisations. L'exploration de texte cherche des réponses à des questions difficiles ou impossibles à résoudre avec les moteurs de recherche seuls. Voici quelques exemples de tels services :

- Résumer les documents décrivant la consommation de produits dans une région particulière .
- Enquêtez sur les plaintes des clients, provoquez des changements dans le comportement des consommateurs, analysez l'image de l'entreprise, etc.
- Gestion de la relation client : Transférez les e-mails des clients reçus sur votre site Web vers le service approprié afin que vos clients puissent répondre le plus rapidement et le plus précisément possible.
- Apprenez à connaître le réseau de relations entre les personnes et les entreprises.

Chacune de ces tâches est un cas particulier du schéma général de la figure [2] ci-dessous.

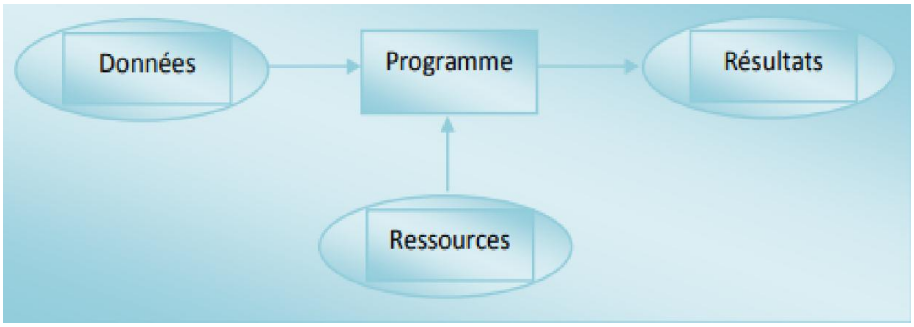


Figure 1.1 Schéma général d'une tâche du Text Mining

1.4. Processus de fouille de texte :

Les étapes requises pour effectuer le processus d'exploration de texte sont les suivantes:

- **Acquisition :** Sources de données telles que des corpus de texte, des bibliothèques numériques et le Web.
- **Filtrage :** Sélectionnez les mots les plus pertinents (technique de sélection d'attributs).
- **Nettoyage des données :** segmentation du texte, suppression des mots vides, lemming.
- **Identification des mots apparentés :** analyse statistique (ngram), analyse sémantique, analyse syntaxique ou analyse structurelle (extraction d'attributs).
- **Extraction de connaissances :** Appliquez l'un des algorithmes d'exploration de texte [4].

1.5.Application d'exploration de texte

L'importance du text mining augmente de jour en jour. Certains domaines importants utilisent des techniques et des outils d'exploration de texte pour trouver des informations pertinentes trouvées dans de grandes quantités de texte dans divers formats.

Au sein de ces domaines, selon [3], on peut citer :

- Rechercher une information.
- Applications biomédicales.
- Filtrage des communications.
- Demande de sécurité.

- Gestion de la propriété intellectuelle.
- Analyse des sentiments.

1.6. Technologie de fouille de texte :

L'exploration de texte est similaire à d'autres domaines hautement complémentaires tels que le traitement automatique du langage (TAL), la recherche de documents (RI) et l'extraction d'informations (EI).

1.6.1. Le traitement automatique du langage "TAL":

Au cours des 15 dernières années, avec la diffusion des ordinateurs et d'Internet, l'application de la PNL dans le domaine de la linguistique a doublé au sens large. La PNL est un domaine à la frontière entre la linguistique et l'informatique, impliquant l'application de programmes et de techniques informatiques à tous les aspects du langage humain.

1.6.2. Rechercher des informations "RI":

La recherche informationnelle « RI » s'intéresse à l'ensemble des documents et aux thématiques qu'ils traitent afin de comparer des documents et de reconnaître des typologies. Tente de détecter tous les thèmes existants.

1.6.3. Extraction d'informations "EI":

L'extraction d'informations consiste à fournir des données exprimées en langage naturel à une base de données structurée. Cela inclut la reconnaissance de mots dans un texte en langage naturel qui correspondent à chaque champ de la base de données. L'analyse est locale. L'extraction d'informations nécessite une analyse syntaxique lexicale et morphologique pour identifier les composants du texte (phrases, mots, verbes, adjectifs) et leurs propriétés afin d'identifier les phrases connexes et d'extraire l'information souhaitée. C’est donc plus compliqué[4] .

1.7.Conclusion

Le text mining est un sujet scientifique interdisciplinaire en croisement entre la fouille de données et le TAL. Dans ce chapitre, nous avons présente le domaine ainsi que les différents concepts et notions en question. Le chapitre suivant focalise sur notre tache qui est la classification de textes.

Chapitre 02 : Classification de textes

2.1 Introduction :

Ce chapitre décrit la classification automatique de texte, ou plus précisément la classification de texte. Voici quelques définitions de la classification et des différents jeux de mots utilisés : classification, catégorisation ou clustering, puis les différents buts et intérêts de la classification, et confluence avec d'autres domaines comme la recherche d'information, puis le texte. Toutes ces étapes expliquent le processus général de classification, et enfin, les problèmes spécifiques au texte dans l'apprentissage automatique.

2.2 Définition de la classification

La classification automatique de documents est un problème connu en informatique, il s`agit d`assigner un document a une plusieurs catégories ou classes. Le problème est différent selon la nature des documents en question, en effet la classification de textes diffère de la classification de documents images, vidéo ou encore son. On peut aussi imaginer des classifications selon des paramètres associés aux documents tels que par exemple l`auteur, la date de parution… Dans le cadre de ce projet et dans la suite de rapport nous nous baserons sur la classification de documents de type texte selon leur contenu.

La classification de textes est une tâche générique qui consiste à regrouper de manière automatisée des documents qui se ressemblent suivant certains critères à savoir les critères observables tels que le type du document, l`année, la discipline, l'édition, etc... Ou s'il faut attribuer une ou plusieurs catégories à un document à partir de critères de contenu et d'une liste prédéfinie. La classification de texte est définie comme une opération qui identifie des classes d'équivalence entre des segments de texte en tenant compte du contenu de l'information (mots, n-gram, etc.).

2.3. Définition formelle

Formellement, la catégorisation de texte CT consiste à associer une valeur booléenne à chaque paire (dj, ci) ∈ D × C, où D est l'ensemble des textes et c'est l'ensemble des catégories selon que dj∈ ci, ou non. Le but de la catégorisation de texte est de construire une procédure (modèle, classifieur) Φ : D × C ->B qui associe une ou plusieurs étiquettes (catégories) à un document dj avec la fonction F: D -> C, la vraie fonction qui retourne pour chaque vecteur dj une valeur ci [7].

2.4.1. Les méthodes de classification automatique

L'objectif de CT est de catégoriser automatiquement les documents dans des catégories qui ont été définies soit préalablement par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore clustering.

2.4.2. Apprentissage non supervisé (Clustering)

En apprentissage non supervisé, les données d'entrée n'ont pas encore été classées. C'est aussi à l'algorithme de découvrir des structures plus ou moins cachées dans les données elles-mêmes et de constituer des groupes d'individus aux caractéristiques communes [11].

La classification non supervisée consiste à retrouver automatiquement une organisation cohérente de groupes de documents similaires pour former un groupe cohérent (classe ou cluster), et les statistiques correspondent au clustering .C'est aussi le concept de recherche d'information. Par conséquent, le clustering consiste à diviser un objet (dans ce cas du texte) en groupes sans connaître à l'avance la classe d'appartenance.

Les techniques pour réaliser de tels regroupements représentent un domaine d'étude très riche, dont l'inventaire dépasse le cadre de ce document. L'apprentissage non supervisé est utilisé dans plusieurs domaines, notamment :

- Médecine : Découverte d'une classe de patients présentant des caractéristiques physiologiques générales.
- Traitement de la parole : créez un système de reconnaissance de la parole humaine.
- Archéologie : Regroupement d'époque d'objets.
- Traitement d'images o Classement des documents. Il existe plusieurs types d'algorithmes d'apprentissage non supervisé dans la littérature, tels que les algorithmes de division et les algorithmes de classification hiérarchique.

• **Partitionnement** : Consiste à regrouper les données selon leur similarité. L'algorithme le plus connu de cette classe est K-means. Il s'agit d'un algorithme qui divise automatiquement le jeu de données en K clusters. Elle consiste à sélectionner d'abord les k points qui représentent le centre du groupe formé, puis à associer les autres points au centre le plus proche. Cette cartographie se fait en calculant la distance entre les points. Plusieurs distances peuvent être définies telles que la distance euclidienne ou la distance de Manhattan. Par la suite nous procédons à une étape de raffinement des groupes de façon itérative, le raffinement se fait par le recalcule des centres des groupes après chaque itération et par une réaffectation des points aux groupes. L`algorithme s`arrête quand aucun point ne bouge. [10]

• **La classification hiérarchique** : il existe deux types de classification hiérarchique : Ascendante et descendent. La classification ascendante consiste à utiliser une matrice de similarité afin de partir d`une répartition fine vers un groupe unique. Donc, il s`agit de fusionner les groupes jusqu`à ce qu`on obtient un seul groupe englobant tous les autres. Cette classification peut être représentée par un arbre hiérarchique ou dendrogramme. La classification descendante se présente comme l`inverse de la classification ascendante. Donc il s`agit de décomposer un cluster unique en sous-groupes jusqu`à l`obtention des singletons. [9]

2.4.3. Apprentissage supervisé (Catégorisation)

Contrairement à l'apprentissage non supervisé, nous commençons ici avec un ensemble de classes pré-connues et définies. Il y a aussi la première sélection de données dont la classification est connue. Ces données sont considérées comme indépendantes et distribuées de manière similaire. Ils nous aident à apprendre l'algorithme. La classification est effectuée par l'algorithme selon le modèle appris. [9]

La classification de texte est le processus d'attribution de texte à une ou plusieurs catégories ou classes prédéfinies. Il adhère à la classification surveillée de l'apprentissage automatique et de l'identification dans les statistiques, mais la recherche d'informations utilise une terminologie (filtrage ou routage) plus proche de votre application.

Ce problème utilise principalement des techniques d'apprentissage automatique et applique de nombreux algorithmes d'apprentissage supervisé (Naive Bayes, KneaRest Neighbors, Decision Trees, Support Vector Machines, Neural Networks, etc.).

2.4.4. Avantages et inconvénients

Les forces et les faiblesses des deux approches sont :

- Les groupes ou clusters acquis par la méthode non supervisée sont d'une qualité et d'une précision supérieure à celles de la méthode non supervisée.
- Utilisez des techniques non supervisées pour voir ce que vous attendez. Cela donne de meilleurs résultats par rapport à l'approche non supervisée.
- L'avantage de l'approche non supervisée est que vous pouvez effectuer des tâches de similarité sans avoir besoin de données expertes.
- L'inconvénient de l'approche contrôlée est qu'il peut être difficile d'obtenir des données d'experts.
- Le principal inconvénient de l'approche non supervisée est qu'elle nécessite l'intervention d'experts lors de la phase d'évaluation des résultats.

2.4.5. Classification Supervisée Vs Classification non Supervisée

La classification Supervisée consiste à identifier la classe à laquelle appartient l'objet en fonction de certaines caractéristiques descriptives. Cette approche permet l'attribution automatique de documents à des classes existantes.

L'objectif est de trouver une relation fonctionnelle, également appelée modèle prédictif, entre le texte classifié et un ensemble de catégories. Pour estimer un modèle prédictif, vous devez disposer d'un ensemble de texte préalablement marqué appelé ensemble d'apprentissage. A partir de cet ensemble, les paramètres du modèle

prédictif le plus efficace possible sont estimés. Minimise la quantité d'erreur qui se produit dans la prédiction.

Contrairement à la classification non supervisée, qui nécessite que l'ordinateur détecte lui-même des groupes de documents, la classification non supervisée suppose que la classification des documents existe déjà. C'est le cas, par exemple, des bibliothèques et des moteurs de recherche. Le but dans ce cas est de classer automatiquement les nouveaux documents. Par conséquent, vous devez d'abord apprendre un modèle ou un classificateur à partir d'un ensemble d'apprentissage composé de paires (objets, classes).

Contrairement à la classification non supervisée, la classification supervisée peut mesurer l'importance de chaque mot pour classer de nouveaux documents. Par exemple, une mesure (gain d'information) calcule la typicité d'un concept. Plus un mot est associé à une catégorie plutôt qu'à une autre, plus il devient important. Si le nouveau document contient le mot, le mot est très discriminant. De nombreuses mesures similaires ont été développées.

Enfin, contrairement à la classification non supervisée, il est ici facile d'évaluer les résultats de la classification. A partir de N exemples de documents classifiés, utilisez une partie du document pour la formation et le reste pour les tests. La phase de test applique un algorithme de classification à chaque document pour voir si le moteur trouve la bonne classe. Bien entendu, les résultats de ce test ne sont pas garantis si la machine doit classer de nouveaux documents. (Vous devez réussir le test, mais ce n'est pas suffisant) [9].

2.5 Algorithmes d'apprentissage

En apprentissage automatique, différents types de classificateurs ont été développés. L'objectif est que chacun atteigne son propre niveau de précision et d'efficacité. Avantages et inconvénients. Cependant, ils partagent des caractéristiques communes.

Une multitude de classificateurs existants pour le regroupement et la différenciation famille élargie. La page suivante détaille quelques algorithmes L'algorithme de classification naïve de Bayes utilisé dans notre étude est d'autres, mais souvent utilisés comme points de référence en raison de leur simplicité.

Il existe de nombreux algorithmes d'apprentissage supervisé, notamment :

- Algorithme K-plus proches voisins (ou KNN)
- Les Arbres de décision.
- machines à support de vecteurs (ou SVM).
- les Réseaux de neurones (RNA).
- Algorithme de Naïve Bayes.

2.6.1. Algorithme des k-voisins les plus proches KNN

2.6.1.1 Définition

L'algorithme des k-voisins les plus proches («k-nearest neighbors» ou kNN) est Méthode d'apprentissage basée sur les instances.

La méthode ne nécessite pas de phase d'apprentissage; c'est l'échantillon d'apprentissage , associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

Lorsqu'un nouveau document arrive à classer, il sera comparé au document de formation Utilisez la similitude. Ensuite, son k-plus proche voisin est considéré. Faites attention à ces catégories, et les catégories les plus fréquentes parmi vos voisins sont attribuées aux documents suivants :

Classer, Il s'agit de la version de base de l'algorithme qui peut être améliorée. Souvent nous Pondère les voisins en fonction de la distance par rapport au nouveau texte.

2.6.1.2. Principe de fonctionnement

L'algorithme KNN se compare à un algorithme qui a déjà été classé en recherchant le K le plus proche voisin. Une fois ceux-ci déterminés, de nouveaux documents seront affectés à la catégorie. Cela inclut le plus grand voisinage de K trouvé. [10]

Deux paramètres sont utilisés : la fonction de similarité à comparer avec le nombre K Nouvelle documentation pour ce qui est déjà classifié, comme la distance euclidienne Il est donné par la formule suivante **:**

$$d(x, y)=\sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

La figure suivante illustre la fonctionnement de l'algorithme KNN :

Paramètre : le nombre K de voisins
Contexte : un échantillon de L textes classés en C = c_1, c_2, ..., c_n classes
Début
  Pour chaque texte T faire
    Transformer le texte T en vecteur T = (x_1, x_2, ..., x_m),
      Déterminer les K plus proches textes du texte T selon une métrique de distance,
    Combiner les classes de ces K exemples en une classe C.
  Fin pour
Fin
Sortie : le texte T associé à la classe C.

La distance entre un texte et ses voisins se fait via une métrique de distance. Cette métrique peut être comme suit :

- **Une mesure de cosinus** qui calcule le produit interne entre deux vecteurs **a** et **b**.Divisez cela par le produit des normes de ces deux vecteurs. Expression les mesures du cosinus sont :

$$Cosinus\ (a, b) = \frac{\sum(a*b)}{\sqrt{\sum a^2 * \sum b^2}}$$

- **Mesure de Distance euclidienne** La formule de la mesure de Distance est comme suivante :

$$D\ (a, b) = \sqrt{\sum |a - b|^2}$$

Mesure de Jaccard La formule de la mesure de Jaccard est :

$$J(a\ ,\ b) = \frac{\sum(a*b)}{\sum a^2 * \sum b^2 \sum ab}$$

2.6.1.3 Critiques de la méthode:

L'avantage de cette méthode est sa simplicité et son efficacité. Méthode populaire ; cependant, on peut lui reprocher d'utiliser des nombres Le nombre d'objets importants pour calculer et classer les similitudes avec de nouveaux objets, et plus le nombre est élevé plus le nombre d'objets est grand, plus le temps d'exécution est important [10].

2.6.1.4 Les domaines d'application :

Cette méthode peut être appliquée dès que vous pouvez définir la distance sur le terrain. Cependant, il est possible de définir des distances dans des domaines complexes tels que l'information. Géographie, texte, images, audio. C'est un critère à choisir de temps en temps En effet, la méthode KPPV est difficile à traiter des données complexes avec d'autres méthodes. Peut en noter également que cette méthode est robuste au bruit [10].

2.6.2. Les arbres de décision :

2.6.2.1. Définition :

Les arbres de décision sont une méthode d'apprentissage plus courante. Ou alors Les algorithmes connus sont ID3 (Quinlan 1986) et C4.5 (Quinlan 1993). Vous aussi populaire dans la classification des documents.

Utilisez les arbres de décision comme toute autre méthode d'apprentissage supervisé. Exemple, Si vous avez besoin de catégoriser vos documents, vous devez créer l'arborescence suivante : Décision par catégorie. Pour déterminer quelle catégorie est nouvelle pour les documents, utilisez l'arbre de décision pour chaque catégorie qui envoie le document Classer. Chaque arbre répond "oui" ou "non" (prend une décision).

Plus précisément, pour chaque nœud de l'arbre de décision, testez (SI ... ALORS)Les valeurs des feuilles sont "oui" ou "non". Chaque test examine la valeur de chaque attribut Exemple. En fait, l'exemple est censé être un ensemble d'attributs/valeurs. Pour certains Dans le document, chaque attribut peut être un mot et la valeur peut être, par exemple, 0 ou 1 selon qu'il s'agit d'un attribut ou non. Si Word appartient au document. [2]

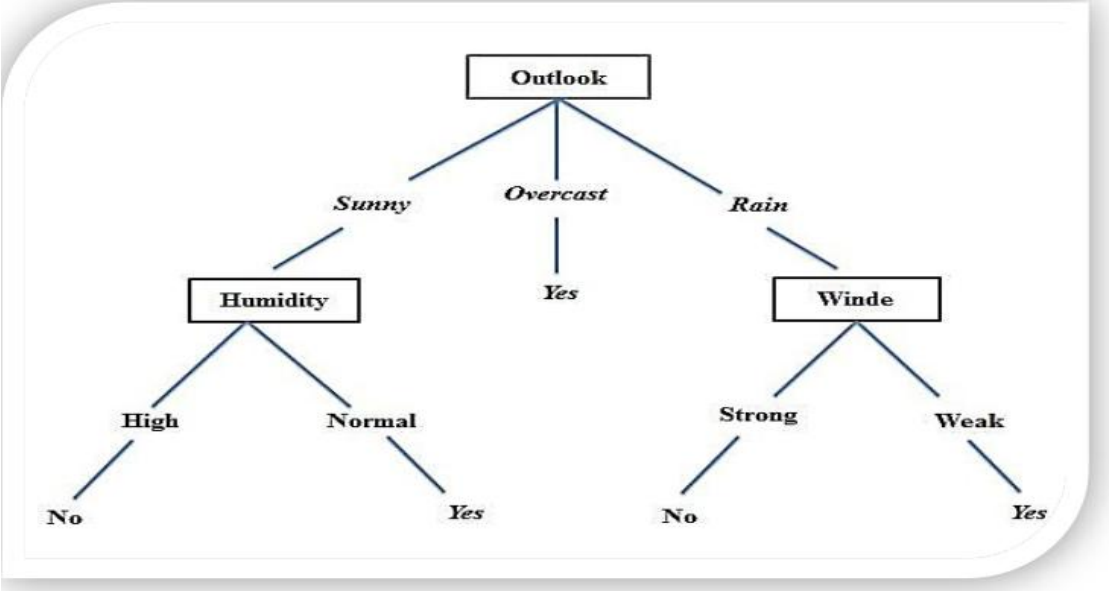


Figure 2.1 : l'arbre de décision

Pour construire un arbre de décision, vous devez connaître les attributs à tester sur chaque nœud. C'est Processus récursif. Utiliser des calculs pour déterminer les attributs à tester à chaque étape Statistiques qui déterminent à quel point cet attribut sépare les exemples oui / non.

Créez ensuite un nœud contenant ce test et créez autant de descendants que la valeur.

Exemple : Lors du test d'existence d'un mot, les valeurs possibles sont existence/absence. Avoir Il y a donc à chaque fois deux descendants par nœud..

2.6.2.2. Algorithme :

En général, l'algorithme de l'arbre de décision se présente de la façon suivante :

```
Arbre ← arbre vide ; nœud_courantracine
Répéter
    Décider si le nœud courant est terminal
        Si le nœud terminal alors lui affecter une classe
        Sinon sélectionner un test et créer autant de nœuds fils qu'il y a de réponse au test
        Passer au nœud suivant (s'il existe)
Jusqu'à obtenir un arbre de décision
```

2.6.2.3 Critiques de la méthode :

Les arbres de décision sont une méthode largement utilisée pour des raisons d'efficacité. Simplicité par rapport aux autres méthodes existantes. En fait, c'est assez compréhensible La règle est de type "Si... Alors...", donc pour tous les utilisateurs. Utilisation simultanée de variables qualitatives et quantitatives (variables discrètes ou continues). Elle le classement est rapide. Suivez un chemin unique pour classer de nouveaux objets. Un arbre de la racine à la feuille qui correspond à cette classe. Cependant, ses performances Les arbres peuvent être très complexes s'il y a beaucoup de classes. Pas toujours optimal. Construire un arbre de décision cela prend généralement beaucoup de temps car il faut trouver la bonne sélection d'attributs. Pour les données La phase d'apprentissage doit être relancée car elle évolue dans le temps. Un échantillon complet qui comprend à la fois des échantillons nouveaux et anciens.

2.6.2.4. Les domaines d'application :

Cette méthode peut être utilisée dans plusieurs domaines tels que : Recherche (Effets pour comprendre les principales priorités lors de l'achat d'un produit Frais publicitaires), revenus (par région, des marques, des vendeurs), analyse des risques (pour identifier les facteurs prédictifs du comportement de non-paiement), soins de santé (pour examiner les relations entre) Maladies spécifiques et caractéristiques physiologiques ou sociologiques) [4].

2.6.3. Machines à support de vecteurs (ou SVM)

Les machines à support de vecteurs (SVM) sont à l'origine de nouvelles méthodes de catégorisations, bien que les premières publications sur le sujet datent des années 60.

Avant d'aborder les principes généraux de fonctionnement des SVM, voici quelques notions de Base:

- **Hyperplan** : Un séparateur pour les objets de la classe. A partir de ce concept, nous pouvons dire Évidemment, vous trouverez de nombreux hyperplans, Le SVM nécessite un hyperplan, qui est la distance minimale à l'exemple de formation.Est le maximum, cet hyperplan est appelé l'hyperplan optimal, et la distance est appelée l'hyperplan optimal.Taux de profit.

- **Vecteurs Support** : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier.

Voici un schéma représentatif de ces notions**[4]** :

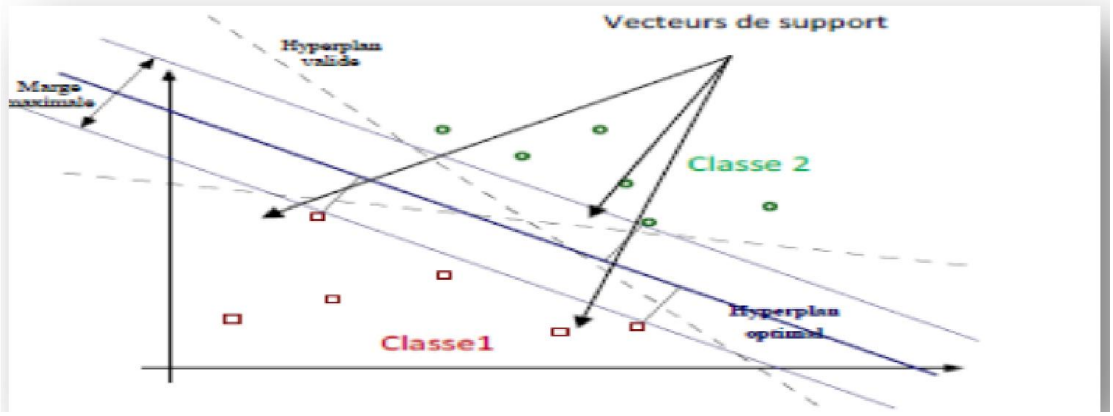


Figure 2.2 Les vecteurs à support

le principe SVM consiste en une stratégie de minimisation du risque structurel. Le problème se résume à trouver la frontière de décision qui divise l'espace en deux zones. Trier les données correctement et trouver l'hyperplan le plus éloigné .De tous les exemples. Je veux maximiser la marge, c'est-à-dire que je veux maximiser la distance du point Le plus proche de l'hyperplan.

Dans la classification de texte, l'entrée est un document et la sortie est C'est une catégorie. Je veux apprendre en examinant un classificateur binaire. Un hyperplan qui sépare les documents qui appartiennent à une catégorie de ceux qui n'y appartiennent pas Partie [10].

Les grandes dimensions de SVM le rendent idéal pour la classification de texte. Il n'y a pas d'impact car il protège contre le surapprentissage. Autrement dit, il insiste Il y a peu d'attributs complètement inutiles pour les tâches de classification et les SVM Vous pouvez éviter les choix agressifs qui peuvent entraîner une perte d'informations. Vous pouvez vous permettre de détenir plus d'attributs. Une autre fonction Lorsqu'il est représenté par un vecteur, le document texte est L'entrée est zéro.

Cependant, SVM convient aux vecteurs dits creux. Un autre aspect positif de Le SVM peut être réglé manuellement, aucun réglage manuel n'est donc nécessaire. Trouver automatiquement les paramètres correspondants [13].

2.6.4. Classification naïve bayésienne

La classification naïve bayésienne est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésienne naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs Linéaires. Un terme plus approprié pour le modèle probabiliste sous-jacent pourrait être « modèle à Caractéristiques statistiquement indépendantes ».

En termes simples, un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classifieur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille. Selon la nature de chaque modèle probabiliste, les classifieurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé.

En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue, [20]. Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats.[21]

L'avantage du classifieur bayésienne naïf est qu'il requiert relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables. En effet, l'hypothèse d'indépendance des variables permet de se contenter de la variance de chacune d'entre elle pour chaque classe, sans avoir à calculer de matrice de covariance.

2.6.4.1 Description du modèle bayésien

Le modèle probabiliste pour un classifieur est le modèle conditionne ($\boldsymbol{C}|\boldsymbol{F1}$, … … . . , $\boldsymbol{F\eta}$) où C est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques$\boldsymbol{F1}$, … … . . , $\boldsymbol{F\eta}$.

Lorsque le nombre de caractéristiques $\boldsymbol{\eta}$ est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible.

Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble. À l'aide du théorème de Bayes, nous écrivons

$$p(C|F_1,\ldots,F_n) = \frac{p(C)\,p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}.$$

En langage courant, cela signifie :

$$\text{postérieure} = \frac{\text{antérieure} \times \text{vraisemblance}}{\text{évidence}}.$$

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de C et les valeurs des caractéristiques $\boldsymbol{Fi}$ sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables. ($\boldsymbol{C}$,, … … . . , $\boldsymbol{F\eta}$) et peut être factorisé de la façon suivante, en utilisant

plusieurs fois la définition de la probabilité conditionnelle :

$$
\begin{aligned}
& p(C, F_1, \ldots, F_n) \\
& \quad = p(C)\ p(F_1, \ldots, F_n|C) \\
& \quad = p(C)\ p(F_1|C)\ p(F_2, \ldots, F_n|C, F_1) \\
& \quad = p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \ldots, F_n|C, F_1, F_2) \\
& \quad = p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ p(F_4, \ldots, F_n|C, F_1, F_2, I \\
& \quad = p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ \ldots p(F_n|C, F_1, F_2, F_3, \ldots
\end{aligned}
$$

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque $\boldsymbol{Fi}$ est indépendant des autres caractéristiques$\boldsymbol{yi} \neq \boldsymbol{j}$ R alors Pour tout$\boldsymbol{i} \neq \boldsymbol{j}$R, par conséquent la probabilité conditionnelle peut s'écrire

$$p(F_i|C, F_j) = p(F_i|C)$$

$$
\begin{aligned}
& p(C, F_1, \ldots, F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots \\
& = p(C) \prod_{i=1}^{n} p(F_i|C).
\end{aligned}
$$

Par conséquent, en tenant compte de l'hypothèse indépendance ci-dessus, la probabilité conditionnelle de la variable de classe C peut être exprimée par où

$$p(C|F_1, \ldots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

où Z (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de $\boldsymbol{F1}$, … … . . , , à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues.

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure P(C) (probabilité a priori de C) et les lois de probabilité indépendantes P(Fi|C). S'il existe K classes pour C et si le modèle pour chaque fonction peut être exprimé selon paramètres, alors le modèle bayésien naïf correspondant dépend de (k − 1) + n r k paramètres.

Dans la pratique, on observe souvent des modèles où K=2 (classification binaire) et r=1 (les caractéristiques sont alors des variables de Bernoulli). Dans ce cas, le nombre total de paramètres du modèle bayésien naïf ainsi décrit est de 2n+1 , avec n le nombre de caractéristiques binaires utilisées pour la classification.

2.6.4.2 Estimation de la valeur des paramètres

Tous les paramètres du modèle (probabilités a priori des classes et lois de probabilités associées aux différentes caractéristiques) peuvent faire l'objet d'une approximation par rapport aux fréquences relatives des classes et caractéristiques dans l'ensemble des données d'entraînement. Il s'agit d'une estimation du maximum de vraisemblance des probabilités. Les probabilités a priori des classes peuvent par exemple être calculées en se basant sur l'hypothèse que les classes sont équiprobables (i.e chaque antérieure = 1 / (nombre de classes)), ou bien en estimant chaque probabilité de classe sur la base de l'ensemble des données d'entraînement (i.e antérieure de C= (nombre d'échantillons de C) / (nombre d'échantillons total)).

Pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question ; sinon, il faut générer des modèles non-paramétriques pour les caractéristiques appartenant à l'ensemble de données d'entraînement. Lorsque l'on travaille avec des caractéristiques qui sont des variables aléatoires continues, on suppose généralement que les lois de probabilités correspondantes sont des lois normales, dont on estimera l'espérance et la variance.

l'espérance, μ, se calcule avec

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Où N est le nombre d'échantillons et xi est la valeur d'un échantillon donné. La variance, σ^2, se calcule avec

$$\sigma^2 = \frac{1}{(N-1)}\sum_{i=1}^{N} (x_i - \mu)^2$$

Si, pour une certaine classe, une certaine caractéristique ne prend jamais une valeur donnée dans l'ensemble de données d'entraînement, alors l'estimation de probabilité basée sur la fréquence aura pour valeur zéro. Cela pose un problème puisque l'on aboutit à l'apparition d'un facteur nul lorsque les probabilités sont multipliées. Par conséquent, on corrige les estimations de probabilités avec des probabilités fixées à l'avance

2 .7 Critères d'évaluation des classificateurs

Diverses façons existantes aujourd'hui ont pour objectif de comparer les décisions prises par le classificateur automatique a celles des experts humains et de calculer un score de performance:

Pour mieux illustrer ces différentes mesures on prend pour point de départ la table de contingence illustrée par le tableau ci-dessous.

L'ensemble des catégories	Document appartenant à la catégorie	Document n'appartenant pas à la catégorie
Document assignés a la catégorie par le classifieur	*a*	*b*
Document rejetés a la catégorie par le classifieur	*c*	*d*

On définit à partir des statistiques de cette table les mesures suivantes :

- 1· Précision (≪précision≫) : a / (a + b), soit le nombre d'assignations correctes sur le nombre total d'assignations.
- 2· Rappel (≪recalé≫) : a / (a + c), soit le nombre d'assignations correctes sur le nombre d'assignations qui auraient dû être faites.
- 3· Exactitude (≪accuracy≫) : (a + d) / (a + b + c + d). ¬ 4· Erreur (≪error≫) : (b + c) / (a + b + c + d).

Comme un document appartient généralement a un petit nombre de catégories sur l'ensemble, un classificateur qui rejetterait tous les documents présenterait seulement un faible taux d'erreur et une exactitude quand même très élevée. Entrainer un classificateur sur la base de l'optimisation d'un de ces deux critères tendrait a créer un programme qui n'accepte aucun document dans sa catégorie. C'est la raison pour laquelle la précision et le rappel sont les mesures les plus rencontrées dans la littérature [18].

2. F-Mesure :

Plusieurs indicateurs ont été crées, mais le plus usuel est la F-mesure qui prenant en compte la valeur relative de la précision et du rappel est calcule comme suit :

$$\text{F\_Mesure} = \frac{(2 * \text{précision} * \text{rappel})}{\text{précision} + \text{rappel}}$$

2.8 Démarche à suivre pour la catégorisation de textes

Effectuez des opérations de classification automatique de texte comme nous le faisons une fois définie, la démarche générale est la suivante : Ainsi, la première étape consiste à intégrer texte que les machines peuvent comprendre et que les algorithmes peuvent utiliser d'apprendre. La classification des documents est la deuxième étape, et cette étape se fait certainement décisif car cela permettra ou non l'apprentissage de la technologie produit une bonne généralisation à partir de couples (document, classe).

Pour améliorer les performances du modèle, évaluez la qualité du classifieur et les résultats fournis par les différents modèles sont comparés en fin de cycle.

La démarche d'une approche standard de classification automatique de textes peut être résumée de la manière suivante :

- Eliminer les caractères de séparation, les signes de ponctuations, les mots vides, etc…
- Les termes restants sont tous des attributs
- Un document devient un vecteur
- Entrainer le modèle de classification à partir des couples (Document, Classe).
- Évaluer les résultats du classifieur. [17]

2.10. Le processus de la catégorisation des textes

Le processus de catégorisation intègre la construction d'un modèle de prédiction qui, en entrée, reçoit un texte et, en sortie, lui associe une ou plusieurs étiquettes. Pour identifier la catégorie ou la classe à laquelle un texte est associé, un ensemble d'étapes d'après [10] est habituellement suivies :

- Prétraitement.
- Définition de descripteur.
- Sélection de descripteur.
- Pondération ou calcul de poids.
- La réduction de la taille du vocabulaire.
- Evaluation du modèle.

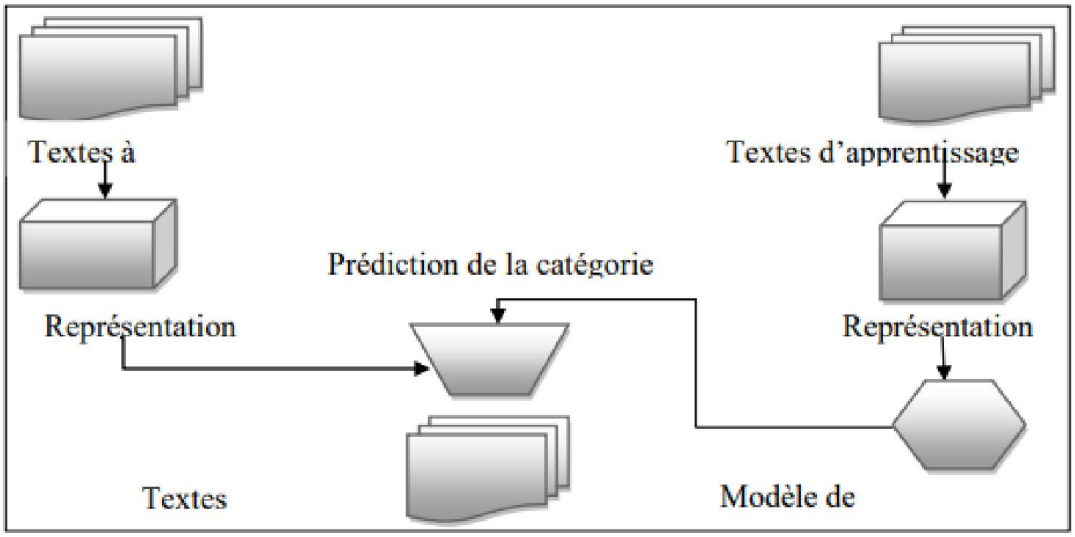


Figure 2.3 Processus de la catégorisation des textes

2.10.1 Prétraitements

Nous aborderons les différentes représentations plus tard document. Ces représentations sont toutes composées de mots, qui sont eux-mêmes une suite de caractères. Il faut donc exécuter avant d'encoder Un document dans un espace de mots, une transformation qui laisse passer l'espace entre caractères et mots.

Le prétraitement du texte est une étape importante dans le processus de classification, car une connaissance inexacte de la population peut entraîner des défaillances opérationnelles. Après la première opération que doit effectuer le système

de classification, c'est-à-dire la reconnaissance des termes utilisés, il est nécessaire de supprimer le maximum d'informations inutiles du document afin que les connaissances stockées soient les plus pertinentes possibles. .. En fait, de nombreux mots dans un document texte fournissent peu (voire pas du tout) d'informations sur le document en question. Des algorithmes dits "mots vides" gèrent leur suppression. Un autre processus, appelé « stemming », peut simplifier le texte tout en augmentant le nombre de caractères utiles, similaire à d'autres méthodes qui suggèrent de supprimer les mots moins importants.

Toutes ces transformations et méthodes font partie du soi-disant prétraitement. Certains d'entre eux sont spécifiques à la langue du document (nous n'effectuons pas le même type de prétraitement sur les documents en anglais qu'en français ou en arabe).

Le prétraitement est généralement effectué en six étapes séquentielles :

- La segmentation
- Suppression des mots fréquents
- Suppression des mots rares
- Le traitement morphologique
- Le traitement syntaxique
- Le traitement sémantique

a. La segmentation

La segmentation est une tâche courante dans le traitement du langage naturel (NLP). Il s'agit d'une étape fondamentale dans toutes les techniques de fouille de texte.

La segmentation est une façon de séparer une pièce de texte en unités plus petites appelées tokens. Les tokens peuvent être des mots, des caractères ou des sous-mots. Par conséquent, la tokénisation peut être classée en trois types : la tokénisation des mots, des caractères et des sous-mots (caractères n-gram).

La manière la plus courante de former des tokens est basée sur l'espace et la ponctuation (., :?...).

Les tokens peuvent être soit des caractères, soit des sous-mots.

Les tokens caractères, exemple : t-e-x-t.

Les tokens sous-mots, exemple : informa-tion

b. Suppression des mots fréquents ou élimination des Mots Outils

Les mots qui apparaissent le plus souvent dans un corpus sont généralement les mots grammaticaux, mots vides (empty words) ou mots outils (stop words) : les articles, les prépositions, les mots de liaisons, les déterminants, les adverbes, les adjectifs indéfinis, les conjonctions, les pronoms et les verbes auxiliaires etc.., qui constituent une grande part des mots d'un texte, mais malheureusement sont

faiblement informatifs, sur le sens d'un texte puisqu'ils sont présents sur l'ensemble des textes.

A titre d'exemple on peut citer en dans la langue arabe, le cas des Prépositions « من », « على », «في » ou de certains mots de liaison «هنا », « الى » etc..

Ces termes très courants peuvent être retirés du corpus pour réduire leur taille. cette la possibilité de réduire la taille des entrées d'index en supprimant les mots vides est décrite comme suit :

Le fait que ces termes soient présents dans presque tous les documents, donc faible sélectivité par rapport aux autres termes.

D'après la loi de Zipf. Leur élimination lors d'un pré-traitement du document permet par la suite de gagner beaucoup de temps lors de la modélisation et l'analyse du document.

Ces mots doivent être supprimés de la représentation des textes pour deux raisons :

• D'un point de vue linguistique, ces mots ne comportent que très peu d'informations. La présence ou l'absence de ces mots n'aident pas à deviner le sens d'un texte. Pour cette raison, ils sont communément appelés « mots vides »

. • d'un point de vue statistique, ces mots se retrouvent sur l'ensemble des textes sans aucune discrimination et ne sont d'aucune aide pour la classification.

Une répartition des mots outils par rapport les mots utiles dans un corpus est représentée dans la figure 2.1.

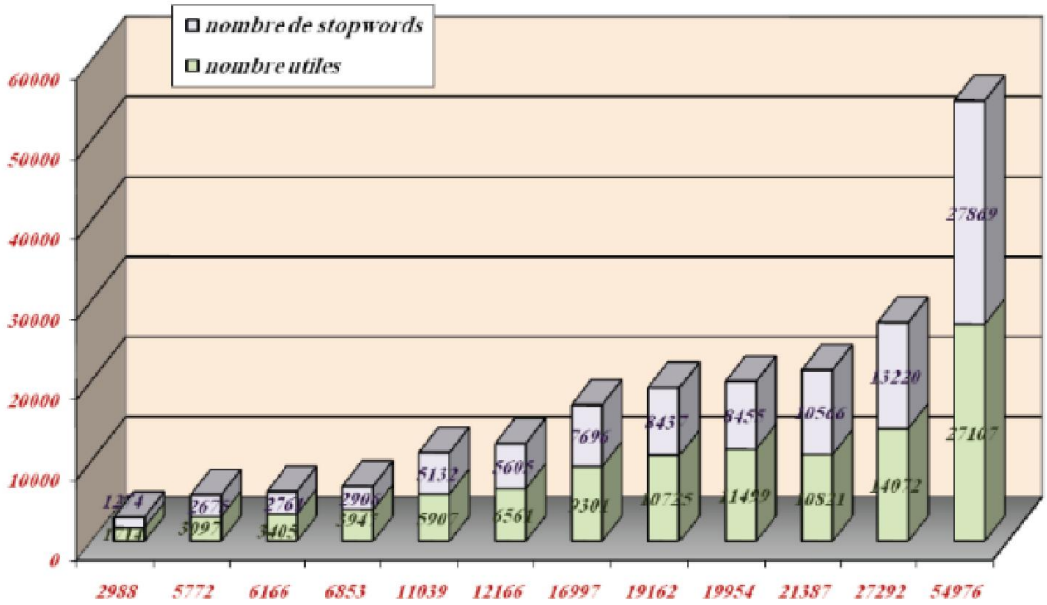


Figure 2.4: Répartition des mots utiles et des mots vides dans un corpus

L'élimination systématique du corpus des mots vides peut se faire par l'intermédiaire d'une liste prédéfinie de mots pour chacune des langues étudiées. Cependant, l'établissement d'une telle liste peut poser des problèmes. D'une part, il

n'est pas facile de déterminer le nombre de mots exacts qu'il faut inclure dans cette liste. D'autre part, cette liste est intimement liée à la langue utilisée et n'est donc pas transposable directement à une autre langue.

c. Suppression des mots rares

En général, les auteurs s'efforcent de supprimer les mots rares qui n'apparaissent qu'une ou deux fois dans le corpus afin de réduire significativement les dimensions des vecteurs utilisés pour représenter le texte. C'est parce que ces mots rares suivent la loi de Zipf. D'un point de vue linguistique, la suppression de ces mots n'est pas toujours justifiée. Certains mots sont très rares, mais très instructifs. Cependant, ces mots sont si peu fréquents qu'ils ne peuvent pas être utilisés dans les méthodes basées sur l'apprentissage. Vous ne pouvez pas créer de statistiques fiables à partir d'un ou deux événements. L'une des méthodes couramment utilisées pour supprimer ces mots consiste à ne considérer que les mots dont la fréquence totale est supérieure à un seuil donné. Enfin, gardez à l'esprit que les mots qui ne contiennent qu'une seule lettre sont généralement ignorés pour les mêmes raisons qu'avant, telles que : Par exemple, le mot "D" en "vitamine D" ou le mot "C" en "langage C".

d. Le traitement morphologique

Il est composé de mots qui ont le même sens en effectuant des traitements au niveau de chaque mot selon les changements morphologiques tels que la flexion, la dérivation et la composition. Ainsi, le but est, par exemple, de regrouper les termes "manger" et "manger", ou "cheval" et "cheval". Parce qu'ils ont le même sens. Cette opération a pour but de réduire la dimension de l'espace de codage du texte pour améliorer encore les performances du système de classement en termes d'espace de stockage et de vitesse de traitement. Il existe plusieurs traitements morphologiques :

- Le stemming ou la désuffixation regroupe sous un même terme (stem) les mots qui ont la même racine. L'extraction des stems se fait par la technique de racinisation (ou stemming) qui utilise à la place des dictionnaires, des algorithmes simples basées sur des règles de remplacement de chaînes de caractères pour supprimer les suffixes les plus utilisés.[8] Le stemming est un traitement linguistique moins approfondie que la lemmatisation, ayant deux avantages : Plu rapide que la lemmatisation (algorithmes simples ne faisant pas référence aux dictionnaires et règles de dérivation) et la possibilité de traiter les mots inconnus sans traitement spécifique.[16] Néanmoins, sa précision et sa qualité sont naturellement inférieures, du fait qu'elle ne gère que les règles principales et ne peut pas prendre en compte les nombreuses exceptions des règles de dérivations. Par exemple, en français l'une des règles préconise de supprimer le « e » final de chaque mot, le mot « fraise » est alors transformé en « frais » ce qui suppose une relation entre les deux mots qui n'existe pas. Qui fait de cette opération dépendante de la langue, nécessitant une adaptation pour chaque langue utilisée.

- La lemmatisation conserve, non pas les mots eux-mêmes, mais leur racine ou lemme. Ce principe permet de prendre en compte les variations flexionnelles (singulier/pluriel, conjugaisons,…) ou dérivationnelles (substantifs, verbes, adjectifs,…) en regroupant sous le même terme tous les mots de la même famille et donc d'améliorer la classification. La lemmatisation est donc une tâche plus compliquée à mettre en œuvre que la recherche de racines, puisqu'elle elle s'appuie sur des outils de TALN, ce qui nécessite beaucoup de ressources linguistiques (dictionnaires, règles de dérivation, etc..). De plus les résultats contiennent encore des erreurs à cause des problèmes de polysémie (ambiguïté) et d'incomplétude des dictionnaires [14].
- Un algorithme efficace, nommé TreeTagger (Schmid, 1994) a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise des arbres de décision pour effectuer l'analyse grammaticale, puis des fichiers de paramètres spécifiques à chaque langue. Toutes les études montrent que les performances des systèmes de classification, après lemmatisation, sont plus nettement supérieures à celles avant lemmatisation.

f. Le traitement syntaxique

La syntaxe traite les combinaisons et l'ordre des mots dans la phrase. Le traitement syntaxique identifie et regroupe un ensemble de mots dont la sémantique dépend de leur association. Par exemple, les mots « casque bleu » ne signifient habituellement pas qu'on a affaire à un casque qui est bleu, mais plutôt à une organisation militaire dépendante de l'ONU. L'analyseur syntaxique a pour but d'identifier ce type de cas. La phase d'analyse syntaxique consiste aussi à éliminer des ambiguïtés comme par exemple les problèmes d'homographie.

g. Le traitement sémantique

Le traitement sémantique consiste à extraire la signification des expressions et traiter la polysémie à savoir les différents sens possibles d'un même mot. Par exemple, cette phase permet de différencier le mot « base » qui peut correspondre à une base militaire ou à une base de données. C'est une opération laborieuse, qui fait appelle aux ontologies, et qui n'est pas aujourd'hui bien maîtrisée et dont l'intérêt en terme de meilleures performances, dans les systèmes de classification, n'est pas toujours démontré.

- Notons, en fin de cette section, que les différents traitements appliqués sur un texte avant sa représentation informatique ne sont pas toujours nécessaire pour toutes les méthodes de représentation d'un texte, notamment le codage en n-grammes, qu'on va étaler par la suite, qui s'en passe d'une bonne partie de ces prétraitements en s'attaquant aux documents, pratiquement, dans leurs états bruts.

2.10.2 Présentation de textes

Les expressions dérivées doivent conserver autant d'informations que possible dans le texte, donc définir ou extraire des caractéristiques dans le texte est une phase importante. Ces propriétés forment les éléments d'information qui composent un document. Le plus petit élément d'information est une lettre. À un niveau supérieur, il y a des mots qui résument un ensemble de lettres, à un niveau plus complet, vous pouvez définir des phrases, des paragraphes, etc., et enfin le document lui-même. Le processus de classification du texte en dépend donc directement et le choix de cet élément de base (descripteur, terme ou fonction) pose problème. Diverses méthodes ont été proposées pour la sélection des termes et la pondération de ces termes. Certains auteurs utilisent des mots comme descripteurs, d'autres utilisent des groupes de mots tels que des mots composés, des expressions et des collocations.

Les sections suivantes définissent les différents types de termes utilisés dans la littérature pour décrire les documents textuels.

2.10.2.1 Représentation en « sac de mots » (bag of words)

Cette méthode consiste à représenter le document sous forme d'un vecteur de mots. Le processus qui permet de convertir le texte d'un document à un ensemble de termes est appelé l'analyse lexicale qui permet de reconnaître les espaces de séparation des mots, les ponctuations, les chiffres,…etc., pour qu'ils seront tous supprimés de la représentation. Cette représentation a comme avantage d'exclure toute analyse grammaticale et toute notion de distance entre les mots, mais présente comme inconvénient la difficulté de délimiter les mots dans certaines langues telles que l'Arabe ou l'Allemand [10].

2.10.2.2. Représentation par phrases

Un certain nombre de chercheurs proposent d'utiliser les phrases comme unité de représentation au lieu des mots comme le cas dans la représentation « sac de mot», puisque les phrases sont plus informatives que les mots seuls, par exemple « recherche d'information », « world wide web », ont un degré plus petit d'ambigüité que les mots constitutifs, et aussi que les phrases ont l'avantage de conserver l'information relative a la position du mot dans la phrase»[3].

2.10.2.3. Représentation avec les racines lexicales

Cette méthode consiste à remplacer les mots du document par leurs racines lexicales, qui peut être réalisée en utilisant l'algorithme de Porter [11] de normalisation de mots qui sert à supprimer les affixes de ces derniers pour obtenir une forme canonique. Cette méthode a comme avantage de regrouper les différentes flexions d'un mot dans une seule composante, et comme inconvénient la perte de sens car la racine extraite peut être commune à des mots se rapportant à des concepts différents. A titre d'exemple : les mots vol, volant, vole ont la même racine vol mais se rendent à trois notions différentes.

2.10.2.4. Représentation avec les lemmes

Cette méthode consiste à remplacer les mots du document par leurs lemmes, elle doit utiliser l'analyse grammaticale afin de remplacer les verbes par leurs formes infinitives et les noms par leurs formes au singulier. En effet, Un mot donné peut avoir différentes formes dans un texte, mais leur sens reste le même. Par exemple, les mots vol, volant et vole seront remplacés par leurs lemmes : vol, volant et voler selon le contexte. Cette représentation est simple mais elle peut causer une perte d'informations donnée par le contexte nécessaire à la distinction des lemmes polysémiques (possèdent plusieurs sens) et la présence de synonymes, considérés comme des lemmes différents même s'ils font référence au même concept [12].

2.11.2.5. Représentation avec les n-grammes

Cette méthode consiste à représenter le document par des n-grammes. Le n-gramme est une séquence de n caractères consécutifs. Cette technique présente plusieurs avantages. Les n grammes capturent automatiquement les racines des mots les plus fréquents sans passer par l'étape de recherche des racines lexicales, indépendante de la langue, les espaces sont pris en considération parce qu'en effet, la non prise en compte de ces derniers introduit du bruit [10].

2.10.2.6. Représentation conceptuelle

Cette méthode consiste à représenter le document sous forme d'un ensemble de concepts, ces concepts peuvent être capturés en utilisant les réseaux sémantiques ou les sous arbres (un sous arbre représente une hiérarchie de concepts). Cette méthode a comme avantage selon [13] de réduire l'espace de travail car les mots qui sont synonymes partagent au moins un concept. Cependant, l'inconvénient majeur de cette représentation est qu'il n'existe pas des bases lexicales pour toutes les langues.

2.10.3 Pondération ou calcul de poids

La pondération des termes permet de mesurer l'importance d'un terme dans un document. Cette importance est souvent calculée à partir de considérations et interprétations statistiques (ou parfois linguistiques). L'objectif est de trouver les termes qui représentent le mieux le contenu d'un document. Les méthodes les plus populaires sont:

2.10.3.1. Le codage TFIDF [10]

- **TF (Term Frequency)** : La fréquence d'un terme est simplement le nombre d'occurrences de ce terme dans le document considéré ;
- **IDF (Inverse Document Frequency)** : La fréquence inverse de document est une mesure de l'importance du terme dans l'ensemble du corpus ;
- **TF*IDF (Term Frequency Inverse Document Frequency):** Le poids d'un terme T dans un document D est calculé comme suit :
  TFIDF (Ti, Dj) = TF (Ti, Dj) * log (N/ DF(T)) (1)
  Avec:
- **TF (Ti, Dj)** : la fréquence du terme dans le document ;

- **N** : le nombre total de documents de la base documentaire ;
- DF(Ti) : le nombre de documents contenant le terme.

2.10.3.2. Le codage TFC

Le codage TF × IDF ne corrige pas la longueur des documents. Pour ce faire, le codage TFC est similaire à celui de TF×IDF mais il corrige les longueurs des textes par la normalisation en cosinus, pour ne pas favoriser les documents les plus longs [12'].

$$\mathbf{TFC\ (ti,\ dj)} = \frac{\mathbf{TF} * \mathbf{IDF(ti,dj)}}{\sqrt{\sum_{S=0}^{|T|} (\mathbf{TF} * \mathbf{IDF(ts,dj)})^2}} \qquad \mathbf{(2)}$$

2.10.3.3. Le codage Lnu

Les différents textes qui composent un corpus ont des tailles différentes dont il faut tenir compte dans le codage des termes. Il existe deux phénomènes à considérer dans les textes longs par rapport aux textes courts [13] :

$$\mathbf{Lnu = L * u;} \quad L = \frac{1+\log(\mathbf{TF(m,t)})}{1+\log(\mathbf{TF(m)})}; \quad U = \frac{1}{0.8+0.2\frac{\mathbf{U(t)}}{\mathbf{U}}} \ \mathbf{(3)}$$

- **TFm** : Fréquence moyenne dans le texte t
- **U(t)** : Nombre de termes uniques dans le texte t
- **U**: Nombre moyen de termes sur l'ensemble des textes du corpus.

2.10.3.4. L'entropie

Une dernière approche de pondération significative s'appuie sur l'utilisation de l'entropie. Cette dernière mesure la dispersion d'un descripteur dans un corpus et peut s'avérer une information importante dans le cadre de la sélection de descripteur et/ou de pondération de la représentation fréquentielle d'un corpus.

$$\mathbf{E(t)} = \sum_d \frac{\mathbf{Ptd}\ \log_2 \mathbf{Pt}d}{\log_2 \mathbf{N}} \quad ; \mathbf{P}\text{td} = \frac{\mathbf{TFt}d}{\mathbf{GF}t} \mathbf{(4)}$$

Où **GF**t représente le nombre total de fois où le descripteur i apparaît dans le corpus de **N** documents.

Une représentation avec l'approche fréquentielle **(TF)** peut alors être la suivante avec pour un terme t et un document **d**:

$$\mathbf{w}\text{td} = \mathbf{(1+E(t))\ log\ (TF}\text{td}\mathbf{+1)} \qquad \mathbf{(5)}$$

2.10.4. Réduction de la taille du vocabulaire

Vu la taille impressionnante des bases textuelles, il est difficile de prendre l'ensemble de tous les mots comme étant des attributs, en effet cela engendre une perte de mémoire et de temps de calcul.

Plusieurs techniques de réduction existent pour réduire la dimension de vocabulaire qui se divise en deux grandes familles :

- **Sélection d'attributs:** qui conserve uniquement les mots utiles à la classification selon un critère fixé préalablement tandis que les autres sont rejetés.
- **Extraction d'attributs**: à partir des attributs de départ, elles créent de nouveaux attributs en faisant soit des regroupements ou des transformations [10].

2.11 Applications de la classification

La classification automatique est une technique utilisée dans plusieurs domaines. Sa capacité prédictive la rend rapide et efficace. Parmi les applications où la classification est utilisée, nous trouvons le filtrage de spam, en effet il s'agit de traiter les messages électroniques textuels, identifier leurs caractéristiques et les classer en deux groupes messages désirés ou non désirés.

Une autre application est la détermination automatique du sujet d'un texte pour le classer automatiquement afin de notifier des personnes intéressées par ce sujet de la présence d'un nouveau texte. [11]

2.12 Quelques problèmes rencontrés dans la catégorisation de textes

Beaucoup difficultés peuvent s'opposer au processus de catégorisation de textes. Des problèmes connus dans la discipline liés à l'apprentissage automatique supervisé comme la subjectivité de la décision prise par les experts, le sur-apprentissage, etc.. mais aussi des problèmes particuliers liés à la nature des données traitées à savoir des données textuelles comme la polysémie, la redondance, Les variations morphologiques ou même L'homographie, etc.. Nous allons signaler les huit principales Dans ce qui suit :

2.12.1. Sur-apprentissage

Le sur-apprentissage s'explique par le fait que le modèle de prédiction n'arrive pas à bien classer les nouveaux textes, pourtant il l'a bien fait dans la phase d'apprentissage en classant correctement les textes de la base d'apprentissage.

Pour limiter le sur-apprentissage, on doit sélectionner des termes pour réduire la dimensionnalité. D'après les expériences antérieures, le nombre de termes doit être limité par rapport au nombre de textes de la base d'apprentissage.

Quelques auteurs recommandent d'utiliser au moins 50 à100 fois plus de textes que de termes. En général le nombre de textes d'apprentissage est limité, c'est pour cela on cherche à agir sur le nombre des termes utilisés en les diminuant, pour éviter ce sur-apprentissage. Sans bien sûr pénaliser le système en supprimant des termes pertinents. [16]

2.12.2. L'homographie

Deux mots sont dits homographes si 'ils s'écrivent de la même façon sans forcément avoir la même prononciation. L'homographie est une sorte d'ambigüité supplémentaire. (Ex : avocat en tant que fruit et avocat en tant que juriste).

2.12.3. Polysémie (Ambiguïté)

Un mot possède, dans différents cas, plus d'un sens et plusieurs définitions lui sont associées. Par conséquent, à cause de la polysémie, les mots seuls sont parfois de mauvais descripteurs ; exemple le mot livre peut désigner une unité monétaire, ou un bouquin.

2.12.4. Les mots composés

Peut-être manque de support pour les mots composés, comme sauver qui peut, comme (Arc en ciel). Leur nombre est très élevé dans toutes les langues, par exemple, traiter le mot (Arc en ciel) comme trois termes distincts peut réduire considérablement les performances du système de classification. Cependant, l'encodage du texte à l'aide de la technologie n gram qui l'affaiblit atténue grandement ce problème avec les mots composés.

2.12.5. La graphie

Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Parce que si un terme est orthographié de deux manières dans le même document (M'sila, m'sila), la simple recherche de ce terme avec une seule forme graphique néglige la présence du même terme sous d'autres graphies, ce qui va influencer les résultats puisque les différentes graphies vont être traitées séparément. Néanmoins du point de vue pratique, le fait qu'un terme inconnu est proche d'un autre terme prouve qu'il a été mal orthographié.

2.12.6. Redondance (Synonymie)

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose. Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. LE FEVRE illustre cette difficulté dans l'exemple du chat et l'oiseau : mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes. [16]

La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat. Lors d'une représentation vectorielle d'un document, ces termes sont représentés séparément, et les occurrences du concept sont dispersées. Il est alors important de rassembler ces terme sen un groupe sémantique commun.

2.12.7. Présence-Absence de termes

La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer, on a donc une relation d'implication entre le mot et le concept associé, quoique on sait très bien qu'il y a plusieurs façons d'exprimer les mêmes choses, dès lors l'absence d'un mot n'implique pas obligatoirement que le concept qui lui est associé est absent du document. Cette réflexion pointue nous amène à être attentifs quant à l'utilisation des techniques d'apprentissage se basant sur l'exclusion d'un mot particulier.

2.12.8. Subjectivité de la décision

Parmi les problèmes classiques usuels dans le domaine de l'apprentissage supervisé c'est la subjectivité de la décision prise par les experts qui décident de la classe à laquelle le texte va être attribué.

Certainement après la lecture du texte à classer, l'expert va trancher à quelle(s) catégorie(s) ce texte appartient en se basant sur le contenu sémantique et le contexte du texte et même en consultant d'autres textes préalablement associés à certaines classes, pour valider la décision prise qui ne peut être que subjective.

Les experts humains ne lisent pas de la même manière ! Ne réfléchissent pas de la même manière ! Donc ne classent pas de la même manière ! Ainsi un même document peut être classé différemment par deux experts, ou encore un même document peut être classé différemment par le même expert, soumis à deux instants différents. [16]

2.13. Conclusion

La classification de texte est devenue un domaine ces dernières années. Axe de recherche pour les entreprises et les particuliers. Cette dynamique est en partie due à la forte demande des utilisateurs pour cette technologie. Il devient de plus en plus indispensable dans de nombreuses situations où la quantité de documents texte électroniques rend impossible le traitement manuel. La classification des textes a considérablement évolué au cours de la dernière décennie, grâce à l'avènement des techniques d'apprentissage automatique qui ont considérablement amélioré le pourcentage de classifications correctes. Dans ce chapitre, nous avons présenté quelques méthodes de classification automatique. Nous avons travaillé sur les méthodes k-Nearest Neighbor (KNN), naïve de Bayes, SVM…

Chapitre 03 : Conception

3.1. Introduction

Le Saint Coran est un texte très complexe, et afin d'atteindre le classement optimal avec ce texte, nous devons adopter plusieurs étapes que nous mentionnerons ci-dessous et passer plusieurs obstacles, en tenant compte du but à atteindre.

3.2. Travaux de recherche liée au Coran

Les recherches qui ont traité des textes du Coran sont peu nombreuses, et la plupart d'entre elles peuvent être dites qu'elles ne sont pas approfondies, car elles traitent de la classification des lettres, et nous mentionnons parmi ces recherches « Processing the Text of the Holy Quran: a Text Mining Study ».Le but de cet article est de trouver une approche pour analyser le texte arabe et ensuite fournir des informations statistiques qui peuvent être utiles aux personnes dans ce domaine de recherche[1].Nous citons une autre étude pour la classification des sourats en Madani ou Makki, l'annotation conceptuelle des textes coraniques et autres tâches[1].

3.3. Les étapes de réalisation

3.3.1. Recueil de versets coraniques :

Cette étape est considérée comme l'étape la plus importante et la plus difficile de notre travail, car il est nécessaire de bien étudier le Coran et de connaître les types qui doivent appartenir à tout, et cela nécessite des connaissances dans ce domaine. , à savoir akida, ahekam et kissas. Le tableau suivant montre le nombre de versets classés :

Classe	Ahekam	Kissas	Akida
Nombre de textes	240	260	370

Tableau 3.1 montre le nombre de versets classés.

3.3.2. Élimination des versets similaires :

Nous avons constaté que la plupart des versets que nous avons classés sont les mêmes. Par conséquent, afin d'obtenir des résultats efficaces, nous avons décidé d'éliminer certains versets similaires, afin que l'utilisation d'algorithmes soit efficace et nous donne le meilleur résultat possible.

3.3.3. Appliquer les algorithmes appropriés :

Après avoir obtenu les versets appropriés, nous devons choisir les algorithmes appropriés pour ce travail, ce qui nous donnera de bons résultats, puisque nous travaillons sur le domaine de la langue arabe.

[1]- http://textminingthequran.com/

3.4. La langue arabe

Dans notre projet, nous devons choisir la langue sur laquelle nous voulons travailler pour la classification, puis nous réfléchissons et étudions. Nous avons décidé de choisir la langue arabe car nous en avons besoin dans plusieurs domaines, et nous ferons une classification appropriée pour celle-ci.

3.5. Démarche proposée

Dans cette partie, nous allons passer en revue un schéma qui met en évidence les étapes les plus importantes du travail :

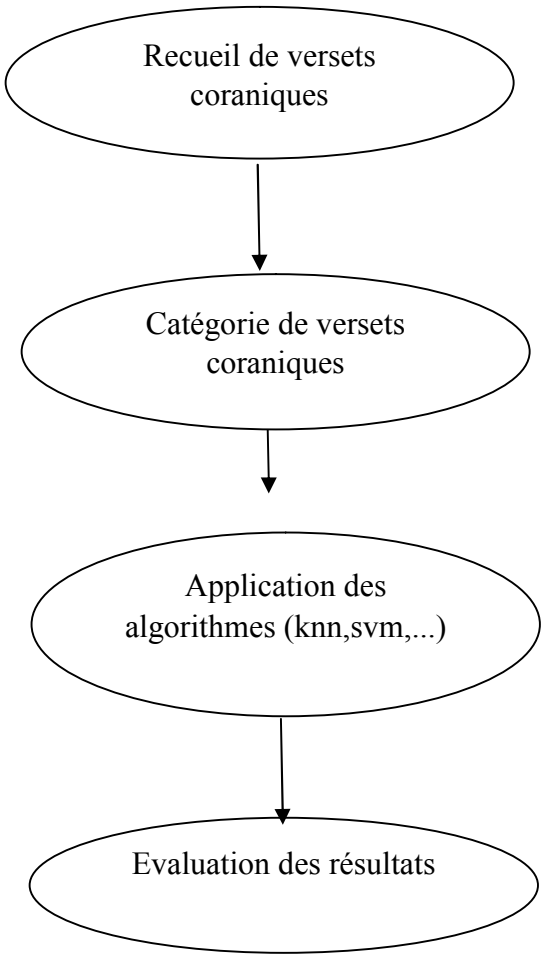


Figure 3.1 .Démarche proposée

3.6. Conclusion :

A travers ce chapitre, nous avons présenté et expliquer les étapes suivis dans notre travail (collecte, choix des algorithmes…), pour réaliser un système de classification automatique pour les versets du Saint Coran.

Chapitre 04 : Implémentation et expérimentation

4.1. Introduction :

Ce chapitre présente notre travail qui consiste à concevoir et mettre en œuvre un système automatisé de classification de textes coraniques. Dans un premier temps, nous présentons les différents outils utilisés : RapidMiner.

4.2. Outils de développement :

Choisir le bon environnement de programmation est Le développement de projets. Cela se produit en raison de plusieurs facteurs. Compilation, simplicité d'utilisation, disponibilité de multiples fonctionnalités, Communication avec d'autres environnements, etc.

Cette plate forme RapidMine a été développé en java, ainsi que le nombre phénoménale des composants et classes mise a la porté des utilisateurs.

4.3. Présentation de la plate forme RapidMiner

RapidMiner est un logiciel libre open source dédié à l'exploration de données. Il contient beaucoup d'outils de traitement de données : lecture de divers formats d'entrée, préparation et nettoyage de données, statistiques, tous les algorithmes de datamining (classification, régression …), évaluations de performances, diverses visualisations.

En plus, il contient les fonctionnalités nécessaires au traitement des données textuelles en arabe, à savoir les techniques de stemming en arabe, les mots vides…

Figure 4.1 RapidMiner

4.4. Historique:

RapidMiner, anciennement connu sous le nom de YALE (Yet Another Learning Environment), a été développé à partir de 2001 par Ralf Klinkenberg, Ingo Mierswa et Simon Fischer de l'unité d'intelligence artificielle de l'Université technique de Dortmund.[23]

À partir de 2006, son développement a été porté par Rapid-I, une société fondée par Ingo Mierswa et Ralf Klinkenberg la même année.[24]

En 2007, le nom du logiciel a été changé de YALE à RapidMiner. En 2013, la société est passée de Rapid-I à RapidMiner.

4.5. Description:

RapidMiner utilise un modèle client/serveur avec le serveur proposé sur site ou dans des infrastructures cloud publiques ou privées.

Selon Bloor Research, RapidMiner fournit 99 % d'une solution analytique avancée grâce à des cadres basés sur des modèles qui accélèrent la livraison et réduisent les erreurs en éliminant presque [la prose du paon] le besoin d'écrire du code. RapidMiner fournit des procédures d'exploration de données et d'apprentissage automatique, notamment : le chargement et la transformation des données (ETL), le prétraitement et la visualisation des données, l'analyse prédictive et la modélisation statistique, l'évaluation et le déploiement. RapidMiner est écrit dans le langage de programmation Java. RapidMiner fournit une interface graphique pour concevoir et exécuter des workflows analytiques. Ces workflows sont appelés "Processus" dans RapidMiner et ils se composent de plusieurs "Opérateurs". Chaque opérateur effectue une seule tâche dans le processus, et la sortie de chaque opérateur constitue l'entrée du suivant. Alternativement, le moteur peut être appelé à partir d'autres programmes ou utilisé comme API. Les fonctions individuelles peuvent être appelées à partir de la ligne de commande. RapidMiner fournit des schémas, des modèles et des algorithmes d'apprentissage et peut être étendu à l'aide de scripts R et Python.[25]

La fonctionnalité RapidMiner peut être étendue avec des plugins supplémentaires qui sont mis à disposition via RapidMiner Marketplace. Le marché RapidMiner fournit une plate-forme permettant aux développeurs de créer des algorithmes d'analyse de données et de les publier dans la communauté.[26]

L'édition gratuite de RapidMiner Studio, qui est limitée à un processeur logique et à 10 000 lignes de données, est disponible sous licence AGPL,[27]

4.6.Caractéristiques principales

- Présentation de l'outil et de ses principes : RapidMiner Studio
- Chargement et préparation des données
  - Chargement des données
  - Analyses descriptives

- graphiques
- Transformation et traitement des données

- Construction de flux
  - Utilisation de modèles prédictifs
  - Amélioration et évaluation des modèles
- Déploiement de modèles
- Aller plus loin : présentation des possibilités en traitement de données textuelles et en traitement de données « big data »

4.7.Présentation du corpus d'expérimentation

Un corpus est un ensemble de documents (textes, images, etc.) qui peuvent être obtenus d'une ou plusieurs disciplines et regroupés pour être traités. Dans la première phase de l'expérience, nous avons utilisé un corpus de poésie coranique. Notre corpus contient 800 textes répartis en trois catégories (Akida, Kisass, Ahkam).

4.8. Le processus de classification a travers RapidMiner :

Pour faire la classification par des méthodes (svm) et mesurer les performances

de classifieur (rappel, précision,…..etc.) par RapidMiner.

Après le lancement du logiciel RapidMiner, on obtient la fenêtre principale :

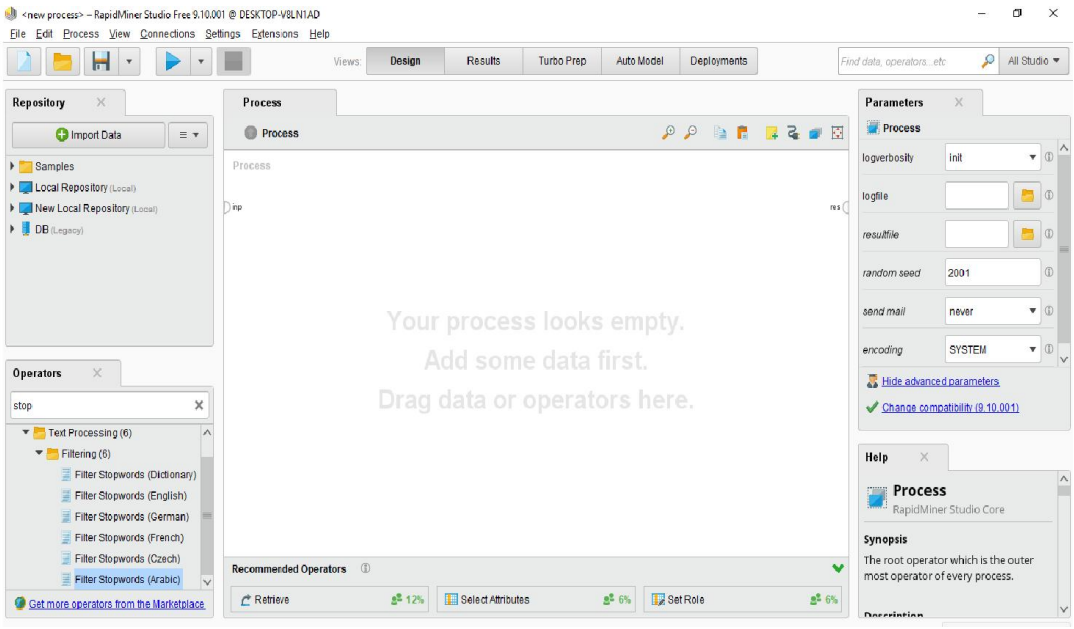


Figure 4.2. Fenêtre principale

Après avoir ouvert l'interface principale,Nous faisons Importation un document texte. Dans la fenêtre des Outils de RapidMiner, allez chercher l'outil Import>Data>Read texte.

On a les paramètres du documents texte :

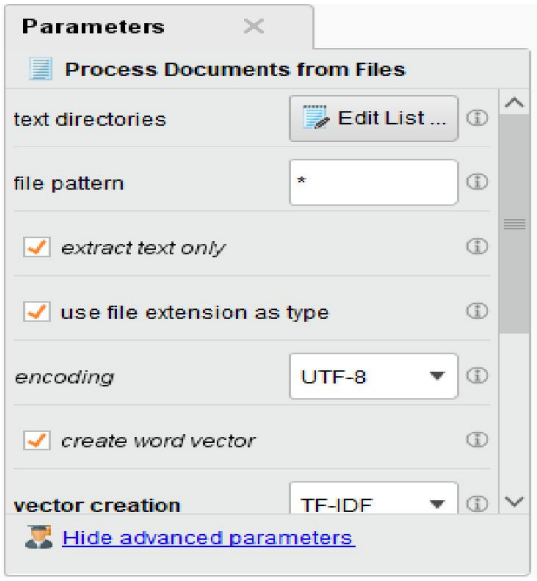


Figure 4.3 : paramètres du document texte

Ona schéma d un processus de RapidMiner :

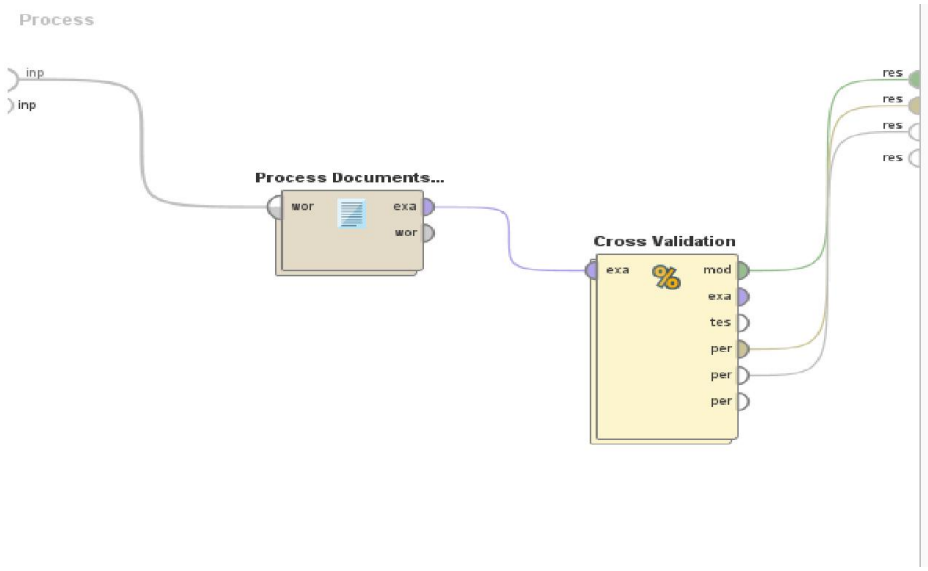


Figure 4.4 : schéma d un processus de RapidMiner

Le processus de la catégorisation des textes tokenize :

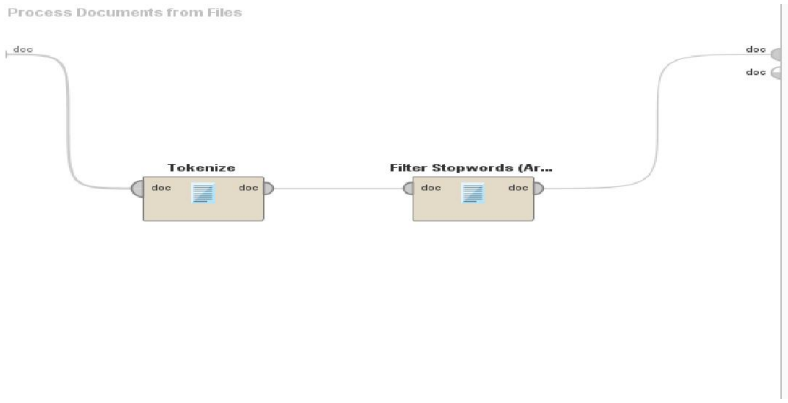


Figure 4.5 : Schéma du processus de la catégorisation des textes tokenize

Pour commencer la classification, nous regardons la méthode Classification

svm fait partie des méthodes de classification de base proposées dans RapidMiner.

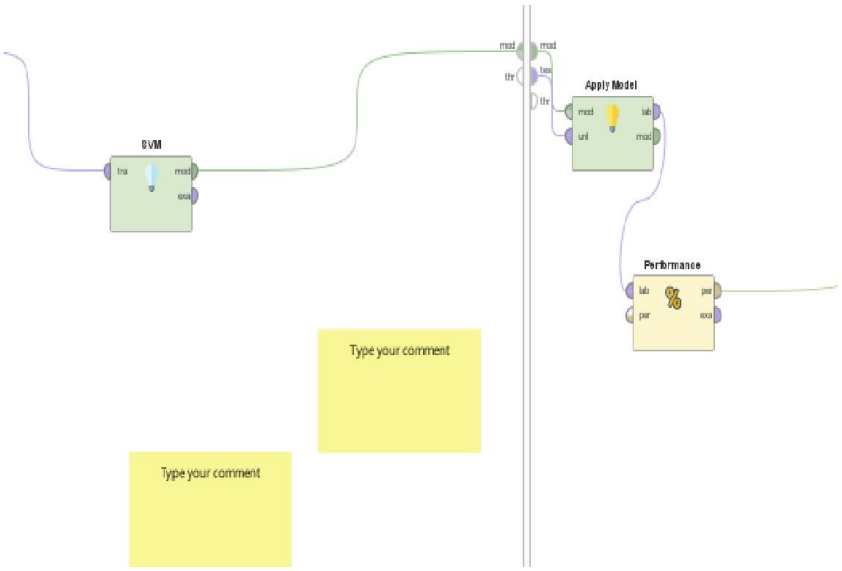
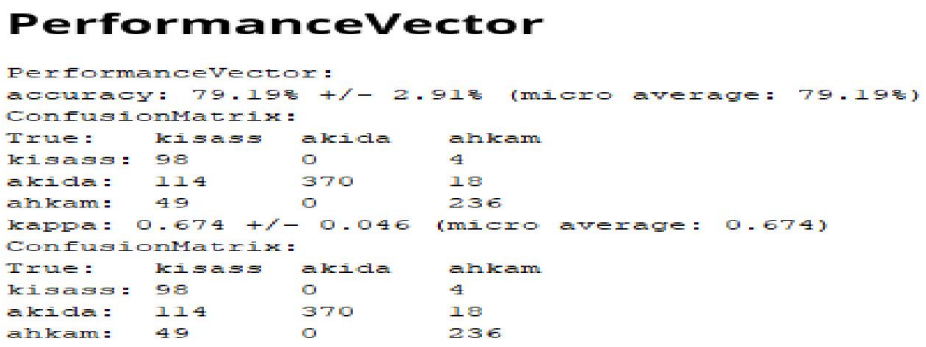


Figure 4.6 : Schéma de l'algorithme de la Classification des textes svm

Une fois que le méthode de classification est choisie on appuie sur le bouton « run», pour lancer l'exécution et prend les résultats. Cette fugure démarquez-vous pour nous les performance le méthode de classification :



PerformanceVector

```
PerformanceVector:
accuracy: 79.19% +/- 2.91% (micro average: 79.19%)
ConfusionMatrix:
True:    kisass   akida    ahkam
kisass:  98       0        4
akida:   114      370      18
ahkam:   49       0        236
kappa: 0.674 +/- 0.046 (micro average: 0.674)
ConfusionMatrix:
True:    kisass   akida    ahkam
kisass:  98       0        4
akida:   114      370      18
ahkam:   49       0        236
```

Figure 4.7 les performance de la méthode de classification

Ce tableau nous montre les résultats d'une propriété tokenize :

accuracy: 87.03% +/- 2.63% (micro average: 87.03%)

	true kisass	true akida	true ahkam	class precision
pred. kisass	148	0	0	100.00%
pred. akida	64	370	0	85.25%
pred. ahkam	49	0	240	83.04%
class recall	56.70%	100.00%	100.00%	

Figure 4.8 Exemple de résultats de classification de base svm (tokenize)

Le tableau 4.1 nous présente les résultats obtenus (en termes de rappel -recall) du processus de la catégorisation des textes en appliquant l'algorithme de classification svm :

Performance	Akida	Ahkam	Kisass	Accuracy
Tokenize	100.00%	100.00%	56.70%	84.81%
Stopwords	100.00%	90.60%	56.68%	79.19%
Stem	100.00%	91.57%	72.44%	79.19%
Stem light	100.00%	93.02%	63.98%	79.19%
3-grams char	100.00%	93.02%	73.56%	**92.35%**
2-grams term	100.00%	91.62%	45.98%	79.19%

Tableau 4.1 Résultats obtenus (rappel) du processus de catégorisation des textes(svm).

Ce tableau nous présente les résultats obtenus du processus de la catégorisation des textes de l'algorithme de la Classification des textes knn :

Performance	Akida	Ahkam	Kissas	Accuracy
Tokenize	40%	82.40%	86.60%	67.03%
Stopwords	39.5%	81.70%	82.02%	57.69%
Stem	33.07%	81.66%	81.77%	56.18%
Sem light	33.10%	81;61%	81.78%	56.18%
3-grams char	92.33%	87.96%	72.44%	**88.18%**
2-grams term	40.02%	81.66%	82.20%	57.69%

Tableau 4.2 Résultats obtenus (rappel) du processus de catégorisation des textes(knn).

Le tableau suivant présente les résultats de la catégorisation des textes de algorithme de la Classification des textes naive Bayes:

Performance	Akida	Ahkam	Kisass	Accuracy
Tokenize	100.00%	93.04%	73.70%	90.66%
Stopwords	12.51%	100.00%	45.68%	42.07%
Stem	12.40%	100.00%	40.44%	43.07%
Stem light	10.55%	100.00%	37.64%	43.07%
3-grams char	100.00%	93.02%	93.56%	**92.35%**
2-grams term	4.06%	100.00%	37.98%	53.07%

Tableau 4.3 Résultats obtenus (rappel) du processus de catégorisation des textes (naive Bayes).

Enfin, nous présentons les résultats obtenus par l'application de l'algorithme decision tree à travers le tableau suivant:

Prformence	Akida	Ahkam	Kisass	Accuracy
Tokenize	100.00%	63.04%	34.70%	74.02%
Stopwords	100.00%	10.43%	9.64%	50.03%
Stem	100.00%	10.43%	9.44%	53.38%
Stem light	100.00%	10.43%	9.64%	53.38%
3-grams char	98.00%	94.02%	44.56%	**81.89%**
2-grams term	100.00%	17.75%	12.06%	53.38%

Tableau 4.4 Résultats obtenus (rappel) du processus de catégorisation des textes (decision tree)

D'après les résultats illustrés dans les tableaux précédents, nous avons pris les remarques suivantes :

La représentation textuelle qui fournit la performance optimale en terme de qualité da classification est n-grams(characters) dans toutes les techniques de classification. En plus, les algorithmes SVM et Naive Bayes sont compétitives et donnent les meilleures valeurs de la métrique accuracy = **92.35%** en termes de résultats de classification. En deuxième place vient KNN , alors que l'algorithme (decision tree) est le plus modeste.

4.10. Conclusion

Dans ce chapitre, nous avons mené une étude afin de classer automatiquement les versets du coran. Pour réaliser notre but, nous avons effectué la chaîne de prétraitement nécessaire pour représenter et modéliser le texte. L'étape suivante consiste à évaluer les performances des techniques ce classification de textes. L'étude expérimentale nous a montré que la meilleure représentation du texte coranique au niveau verset est la suite de caractères (n-grams characters).

La partie expérimentation a indiqué également que les algorithmes les plus performants en terme de qualité de classification sont SVM et Naive Bayes, alors que le plus modeste est (decision tree).

Conclusion générale et perspectives

Dans ce travail de master, nous avons mené une étude pour classer les versets coraniques en appliquant différentes techniques de classification coulées avec diverses méthodes de représentation et traitement de textes (filtrage, stemming, n-grams…). Le but de la classification est d'apprendre à la machine à classer le texte du Coran dans la bonne catégorie en fonction du contenu (akkida, ahkam, kissas).

Nous avons choisi une approche expérimentale, la classification thématique, qui consiste à classer les textes coraniques en fonction de leur contenu à l'aide d'un classificateur. Le but de cette étape est de classer les versets coraniques et d'associer chaque verset à sa catégorie (akida, ahkame, kissas).

L'analyse des résultats obtenus a révélé que le taux de classement est acceptable, mais cela n'empêche pas qu'il y ait quelques problèmes. La partie expérimentation a indiqué que SVM et Naive Bayes sont les algorithmes les plus efficaces parmi les techniques testées. D'après les tests toujours, la représentation de textes par les n-grams de caractères est la plus performante.

On a rencontré aussi quelques problèmes dans les phases de traitement des versets, notamment la similitude des versets dans les significations les versets qui appartiennent à plusieurs classes.

Malheureusement, le temps alloué à ce travail est si court qu'il est difficile d'y remédier

- Elargir notre corpus coranique en intégrant autres versets.
- Appliquer d'autres méthodes à la représentation textuelle, à savoir : Concepts et les méthodes word embedding…
- Intégrer les méthodes de réduction et de sélection des attributs pour évaluer leurs effets dans le processus de classification.
- Etudier autres paramètres pour les algorithmes testés.
- Tester autres classificateurs (réseaux de neurones, apprentissage approfondi…).

Bibliographie

[1] K. Arai, « International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2 »

University saga, 2015.

[2] R. Lefébure, G.Venturi, « *Le Data Mining* » Edition EYROLLES, deuxième tirage 1998.
[3] Taibi, H.LAZREG, «Utilisation des algorithmes d'apprentissage dans la catégorisation automatique thématique de documents Etude de cas : les algorithmes K_PPV, Naïve Bayes», Mémoire de Licence, Université de M'sila, 2011-2012.
[4] R. Saeed, « L'Apprentissage Artificiel pour la Fouille de Données Multilingues: Application à la Classification Automatique des Documents Arabes », Thèse de doctorat en Sciences de l'Information et de la Communication, Université Lumière Lyon 2, 2010.
[5] B. Sadik « Analyse de Données Textuelles pour la Classification Automatique par les Techniques de Text Mining, application à la Langue Arabe », Mémoire de Magister En Informatique, Université de Sétif, 2007.
[6] J. Radwan, « Apprentissage automatique et catégorisation de textes multilingues », Thèse de doctorat, Université Lumière Lyon 2, France, Juin 2003.
[8] B.Ameni « Catégorisation automatique de news à l'aide de techniques d'apprentissage supervisé ». Rapport de Projet de Fin d'Etudes, Université Nice Sophia Antipolis.
[9] A. Soumia, «Processus de classification supervisée de textes arabes par la méthode K PPV Application aux articles de presse», Mémoire de Master, Université de M'sila, 2011-2012 .
[10] T.DERDRA Amel, B. Fatima zahra, « La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue », Mémoire de Master, Université Abou Bakr Belkaid– Tlemcen, 2011-2012.
[11] P. Martin, «An algorithm for suffix stripping », Program, pp 130–137, Morgan Kaufmann Publishers Inc, 1980.
[12] I. Camelia « Représentation de textes a l'aide d'étiquettes sémantiques dans le cadre de la classification automatique », Européen Commission, IPSC, Strasbourg, France,2007.
[13] R. Simon, « Catégorisation automatique de textes et Cooccurrence de mots provenant de documents non étiquetés », Mémoire, Université Laval Québec, Canada, Janvier 2005.
[14] F.Sebastiani.« Machine learning in automated text categorization ». 2002.
[15] L. Phillippe « La recherche d'information - du texte intégral au thésaurus » 2000.
[16] C. Jeremy, Z. Djamel « Une technique de réétiquetage dans un contexte de catégorisation de textes » 2004.
[17] M. Hocine. « Classification Automatique de Textes Approche Orientée Agent ». Mémoire de magister, Département de l'informatique, université d'Aboubekr BelkaidTlemcen. Février 2011.
[18] R. Simon, « Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés » Mémoire présenté à la Faculté des études supérieures de l'Université Laval, Québec, Janvier 2005.
[19] S. Sébastian, « Approches textuelles pour la catégorisation et la recherche de documents manuscrits en-ligne » université de Nantes thèse de doctorat soutenu le 24 mars 2010.
[20] Z. Harry, "The Optimality of Naive Bayes". Conférence FLAIRS 2004.
[21] R.Caruana, A. Niculescu-Mizil: "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international conference on Machine learning, 2006.

[22] N. Eric, M. Serge« CLASSIFICATION BAYESIENNE NAÏVE DE TEXTES », Faculté Polytechnique de Mons, 5ième Electricité, Certi_cat Applicatifs Multimédia.

[23] Guido Deutsch, "RapidMiner from Rapid-I at CeBIT 2010," *Data Mining Blog*, March 18, 2010.

[24] "Interview with RapidMiner's Ingo Mierswa, Ralf Klinkenberg", *KDnuggets*, February, 2010.

[25] "German Predictive Analytics Startup Rapid-I Rebrands As RapidMiner", *TechCrunch*, November 4, 2013.

[26] N. David, "RapidMiner - a potential game changer," Bloor Research, November 13, 2013.

[27] "Interview with Rapid-I Ingo Mierswa and Simon Fischer," *KDnuggets*, August 2011.

[28] RapidMiner Embraces its Community and Open Source Culture Delivering Get-More-Open-Core Predictive Analytics, September 1, 2015.

Webographie :
[7] http:// www.cuy.be/html/typoweb/chap1.html consulté le l'avril 2014.
[20] (http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf)
[21] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.5901&rep=rep1&type=pdf

ملخص

يعد التنقيب عن النصوص من التقنيات المهمة جدًا هذه الأيام و أكثرها طلبا مما توفره من سهولة لمعالجة البيانات النصية الكبيرة بطرق سريعة.هدف الدراسة هو تصميم و انجاز تطبيق يسمح بتصنيف آيات القران الكريم حسب معانيها المختلفة إلى أصناف (قصص ، أحكام ...) باستعمال خوارزميات التصنيف الآلي. نتائج الدراسة التجريبية أظهرت أحسن تمثيل للآية هو n-grams car و أحسن خوارزم للتصنيف هو SVM و Naive Bayes.

الكلمات المفتاحية:التنقيب عن النصوص ، تصنيف النصوص ، القرآن.

Résumé

La fouille de textes est l'une des techniques les plus importantes de nos jours et la plus demandée, car elle permet de traiter rapidement des données textuelles volumineuses. Le but de l'étude est de concevoir et de réaliser une application qui permet la classification des versets du Saint Coran selon ses contenus (KASSAS,AHKAM…) en à l'aide des algorithmes de classification automatique. Les résultats de l'étude expérimentale ont montré que la meilleure représentation de texte est (n-grams car) et le meilleur algorithme de classification est SVMet Naive Bayes.

Mots clé : fouille de textes, classification de textes, Coran.

Abstract

Text mining is one of the most important techniques nowadays and the most demanded, because it allows to process quickly voluminous textual data. The aim of the study is to design and implement an application that allows the classification of the verses of the Holy Quran according to its contents (KASSAS, AHKAM...) using automatic classification algorithms. The results of the experimental study showed that the best text representation is (n-grams car) and the best classification algorithm is SVM and Naive Bayes.

Keywords: text mining, classification,Quran.