UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES
ET INFORMATIQUE APPLIQUÉES

PAR
LABIAD ALI

SÉLECTION DES MOTS CLÉS BASÉE SUR LA CLASSIFICATION ET L'EXTRACTION DES RÈGLES D'ASSOCIATION

JUIN 2017

Résumé

Ce mémoire traite de la problématique de l'extraction automatique des mots clés. Ceux-ci fournissent des informations importantes sur le contenu du document. Ils peuvent aider les utilisateurs à rechercher de l'information d'une manière plus efficace voire mieux ciblée. Ils peuvent également être utilisés dans la création d'index automatique pour une collection de documents ou bien pour une représentation de documents dans les tâches de catégorisation ou de classification.

Dans ce mémoire, nous proposons une nouvelle méthode combinant à la fois la classification, le schéma TF-IDF (Term Frequency-Inverse Document Frequency) et les règles d'association régulières. En effet, notre méthode est une chaîne de traitement qui se déroule en trois phases. D'abord, un classifieur reçoit en entrée la matrice termes-documents produite par le schéma de la pondération TF-IDF dans le but de classifier ses vecteurs de termes. Ensuite, on applique l'algorithme Apriori sur chacune des classes résultantes pour extraire les règles d'association régulières. Enfin, on extrait les mots clés à partir de l'ensemble des règles obtenues. Les résultats d'expérimentation sur des données réelles montrent que notre méthode de sélection de mots clés permet d'obtenir un ensemble de mots clés petit et satisfaisant.

Remerciements

Nous rendons grâce à Dieu qui nous a donné l'aide, la patience et le courage pour accomplir ce travail.

Je tiens à adresser mes plus vifs remerciements à Mr Ismail Biskri, pour m'avoir encadré et pour les recommandations qu'il m'a prodiguées et qui m'ont été d'un grand apport.

Je tiens à adresser mes plus vifs remerciements aux membres de jury, pour avoir accepté d'évaluer mon travail.

Je tiens à remercier ma famille, surtout ma femme, Fatima, pour leurs soutiens et leurs encouragements.

Merci à tous ceux qui ont contribué à l'élaboration de ce travail de près ou de loin et qui méritent d'y trouver leur nom.

TABLE DES MATIERES

Liste des figures

Liste des tableaux

Liste des algorithmes

Chapitre 1

INTRODUCTION

Au cours de ces dernières années, le forage de texte (Text mining) a gagné l'attention de plusieurs chercheurs en raison des quantités importantes de données textuelles qui sont créées dans une variété de réseaux sociaux, le Web et d'autres applications centrées sur l'information. Le Web, par exemple, encourage particulièrement la création d'une grande quantité de contenu textuel par différents utilisateurs dans une forme qui est facile à stocker et à traiter. Ces immenses données textuelles ont créé un besoin de concevoir des algorithmiques qui peuvent apprendre à partir de données d'une manière dynamique et évolutive.

Les données textuelles sont généralement gérées via un moteur de recherche en raison du manque de structures. Un moteur de recherche permet à un utilisateur de trouver des informations utiles à partir d'une collection de documents avec une requête de mots clés (Keywords). L'extraction des mots clés (KE) est chargée de l'identification automatique d'un ensemble d'unités lexicales qui décrivent le mieux le thème d'un document. L'extraction des mots clés est un problème important dans le forage de texte (Text Mining), la recherche d'informations (Information Retrieval) et le traitement du langage naturel (Natural Language Processing). En d'autres termes, les mots clés pertinents extraits peuvent être utilisés dans la création d'index automatique pour une collection de documents ou bien pour une représentation de documents dans les tâches de catégorisation ou de classification.

L'extraction automatique de mots clés a suscité l'intérêt des chercheurs ces dernières années. Nous présentons dans ce mémoire une nouvelle méthode d'extraction

des mots clés basée sur la classification mathématique du texte et l'extraction des règles d'association. Nous pensons que l'extraction des règles d'association à partir de classes résultantes s'avère très efficace dans la sélection des mots clés. Un document textuel peut être représenté sous la forme d'une matrice de taille nxd, où n est le nombre de documents et d la taille du vocabulaire. La $(i,j)^{ieme}$ entrée de cette matrice représente la fréquence normalisée de i^{ieme} mot dans le j^{ieme} document. Au contraire des autres travaux qui font une classification sur les documents, nous effectuerons une classification sur les termes.

Ce mémoire, est constitué de sept chapitres. Dans un premier temps, dans le chapitre 1 nous présentons la problématique et les objectifs de ce projet de recherche.

En deuxième partie, dans le chapitre 2, nous exposons quelques méthodes de classification de documents textuels. Ces méthodes se divisent en deux catégories. La première catégorie concerne les méthodes supervisées qui consistent à classer les documents à partir d'une base de données d'apprentissage contenant des exemples préclassés. La deuxième catégorie concerne les méthodes de classification non supervisées qui permettent de classer les documents sans utilisation d'une base de données d'apprentissage.

Le chapitre 3 traite du modèle vectoriel qui est une représentation mathématique du contenu d'un document ainsi que du schéma de pondération TF-IDF.

Dans le chapitre 4, nous exposons la formulation mathématique des règles d'association régulières, la méthode Apriori utilisée pour l'extraction de ces règles ainsi que les règles d'association avec contraintes qui sont un type particulier des règles d'associations.

Le chapitre 5 présentera notre projet et la méthodologie suivie pour l'extraction des mots clés.

Dans le chapitre 6, nous exposerons la présentation et la discussion des différents résultats obtenus lors de la phase d'expérimentation. Les tests sont effectués sur un livre complet, à savoir « La civilisation des Arabes ».

Enfin, le chapitre 7 donne une conclusion générale, qui est une vue globale et synthétique de notre travail.

Chapitre 2

Les méthodes de classification

2.1 Introduction:

La classification c'est l'action de ranger par classes, par catégories des objets avec des propriétés communes. Il existe deux catégories de classification : classification supervisée et classification non supervisée. Dans la première catégorie, les méthodes consistent à classer les objets à partir d'une base de données dite d'apprentissage, tandis que dans l'autre les méthodes classent les objets sans besoin de cette base de données.

Dans la littérature scientifique, plusieurs méthodes de classification ont été présentées. Dans ce chapitre nous allons présenter les plus connues.

2.2 K-moyen (K-mean) :

La méthode k-moyen a été introduite par MacQueen en 1967. En 1965 Forgy a publié un algorithme similaire, c'est pour cette raison qu'elle est parfois référée à Forgy. C'est une méthode efficace qui permet de diviser un ensemble de données en k classes homogènes.[1]

La méthode K-moyen est utilisée dans l'apprentissage non supervisé. Elle est itérative c'est-à-dire qu'elle converge vers une solution quelque soit son point de départ.[1, 2]

Dans un premier temps, l'algorithme choisit le centre des k clusters à k objets. Par la suite, on calcule la distance entre les objets et les k centres et on affecte les objets aux centres les plus proches. Ensuite, les centres sont redéfinis à partir des objets qui ont été affectés aux différents clusters. Puis, les objets sont assignés en fonction de leur distance aux nouveaux centres et ainsi de suite. L'algorithme se répète donc jusqu'à ce qu'il y ait convergence .[1, 2]

Les k clusters sont produits de façon à minimiser la fonction objective suivante :[3]

$$E = \sum_{r=1}^{k} \sum_{X_i \in C_r} (X_i - g_r)^2$$

Où :

C_r , l'ensemble des classes.

X_i , Un individu appartenant à une classe C_r .

g_r , Le point moyen de la classe C_r .

Il existe plusieurs types de distance pour calculer la distance entre les objets et les k centres. Parmi ces distance, on peut citer : [4]

- La distance Euclidienne.
- La distance de Minkowsky.
- La distance de Manhattan.

La distance Euclidienne est souvent utilisée. Elle est donnée par la formule suivante :[4]

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

Où : x , y sont des vecteurs.

L'algorithme K-moyen est un algorithme glouton (Greedy Algorithm) dont la performance dépend fortement de l'estimation initiale de la partition. Les méthodes d'initialisation couramment utilisées sont Forgy et la partition aléatoire. Dans la méthode de Forgy les k centres initiaux sont choisis au hasard de l'ensemble de données. La méthode de partition aléatoire affecte chaque objet à un cluster aléatoire puis elle calcule le centre initial de chaque cluster.[1, 5]

2.2.1 L'algorithme de K-moyen :

Le fonctionnement de K-moyen se résume dans les étapes suivantes :[3]

1. On choisit k objets au hasard qu'on considère comme des centres pour les classes initiales.

2. On affecte chaque objet au centre le plus proche pour obtenir une partition de k classes.

3. On recalcule les centres de chaque classe.

4. La répétition des étapes 2 et 3 jusqu'à la stabilité des centres.

Algorithme 2.1 : L'algorithme K-moyen

La complexité de l'algorithme du K-moyen est de $O(lkn)$.[3]

Où :

l : est le nombre d'itérations.

k : le nombre des classes telles que $k < n$.

2.2.2 Avantages du K-moyen :

La méthode du K-moyen comporte des avantages, comme par exemple :

1. L'assimilation de cette méthode est rapide.

2. La méthode simple à appliquer.

3. Son adaptation à de larges bases de données. [3]

2.2.3 Inconvénients du K-moyen :

La méthode du K-moyen comporte les inconvénients suivants :

1. La difficulté de comparer la qualité des clusters obtenus. [1]

2. La performance de l'algorithme dépend fortement de l'estimation initiale de la partition. [1]

3. Le choix du paramètre k influence les résultats. [3]

2.3 K plus proches voisins (KNN) :

La méthode des k plus proches voisins (k-nearest neighbor, KNN) est une méthode supervisée. Elle a été utilisée dans l'estimation statistique et la reconnaissance des modèles comme une technique non paramétrique, cela signifie qu'elle ne fait aucune hypothèse sur la distribution des données.[6]

L'algorithme KNN est l'un des plus simples de tous les algorithmes d'apprentissage automatique. Il est un type d'apprentissage basé sur l'apprentissage paresseux (lazy learning). En d'autres termes, il n'y a pas de phase d'entraînement explicite ou très minime. Cela signifie que la phase d'entraînement est assez rapide.[6]

La méthode KNN suppose que les données se trouvent dans un espace de caractéristiques. Cela signifie que les points de données sont dans un espace métrique. Les données peuvent être des scalaires ou même des vecteurs multidimensionnels.[6, 7]

La méthode des k plus proches voisins est utilisée pour la classification et la régression. Dans les deux cas, l'entrée se compose des k données d'entraînement les plus proches dans l'espace de caractéristiques. [6, 7]

Pour trouver la classe d'un nouveau cas, cet algorithme se base sur le principe suivant : il cherche les k plus proches voisins de ce nouveau cas, ensuite, il choisit parmi les candidats trouvés le résultat le plus proche et le plus fréquent.[3, 8]

Pour affecter un nouvel individu à une classe, l'algorithme cherche les k plus proches voisins parmi les individus déjà classés. Ainsi, l'individu est affecté à la classe qui contient le plus d'individus parmi les candidats trouvés. [3, 8]

Cette méthode utilise principalement deux paramètres : une fonction de similarité pour comparer les individus dans l'espace de caractéristiques et le nombre k qui décide combien de voisins influencent la classification.[3]

Pour tester la similarité entre deux vecteurs, la distance est utilisée. Elle permet de mesurer le degré de différence entre deux vecteurs.Il existe plusieurs types de distance parmi lesquels on trouve : [4]

- La distance Euclidienne :

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

Où : x , y sont des vecteurs.

- La distance de Minkowsky :

$$d(x,y)=\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{1/p}$$

Où : x , y sont des vecteurs.

p : paramètre

- La distance de Manhattan :

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i|$$

Où : x , y sont des vecteurs.

2.3.1 L'algorithme des k plus proches voisins :

Soit :

X **:** ensemble d'entrainement**.**

Y **:** étiquettes de classe de X

x : individu inconnu

Pour $i = 1$ **a** m **faire**

Calculer la distance d (X_i, x)

Fin pour

Construire l'ensemble I contenant des indices pour k plus petite distance d (X_i, x)

Retourner Étiquette majoritaire pour $\{Y_i, ou\ i \in I\}$

Algorithme 2.2 : L'algorithme des k plus proches voisins **[9]**

2.3.2 Avantages de la méthode des k plus proches voisins :

La méthode des k plus proches voisins représente des avantages tels que :

1. L'algorithme KNN est robuste envers des données bruitées.[8]

2. La méthode des k plus proches voisins est efficace si les données sont larges et incomplètes.[9]

3. Cette méthode est l'une des plus simples de tous les algorithmes d'apprentissage automatique.[6]

2.3.3 Inconvénients de la méthode des k plus proches voisins :

La méthode des k plus proches voisins comporte des inconvénients tels que :

1. Le besoin de déterminer la valeur du nombre des plus proches voisins (le paramètre k).[7]

2. Le temps de prédiction est très long puisqu'on doit calculer la distance de tous les exemples.[8]

3. Cette méthode est gourmande en espace mémoire car elle utilise une grande capacité de stockage pour le traitement des corpus.[3]

2.4 Les machines à support de vecteurs (SVM) :

Introduit par Vapnik en 1990, les machines à vecteurs de support sont des techniques d'apprentissage supervisé destinées à résoudre des problèmes de classification et de régression. Elles reposent sur deux notions principales : la notion de marge maximale et la notion de fonction noyau.[3, 10]

Les machines à support de vecteurs peuvent être utilisées pour résoudre divers problèmes en bio-informatique, recherche d'information et vision par ordinateur, etc.[10]

2.4.1 Le principe des SVM :

Le but des SVM est de trouver un séparateur entre deux classes qui soit au maximum éloigné de n'importe quel point des données d'entraînement. Si on arrive à trouver un séparateur linéaire c'est-à-dire qu'il existe un hyperplan séparateur alors le problème est dit linéairement séparable sinon il n'est pas linéairement séparable et il n'existe pas un hyperplan séparateur. [3, 10]

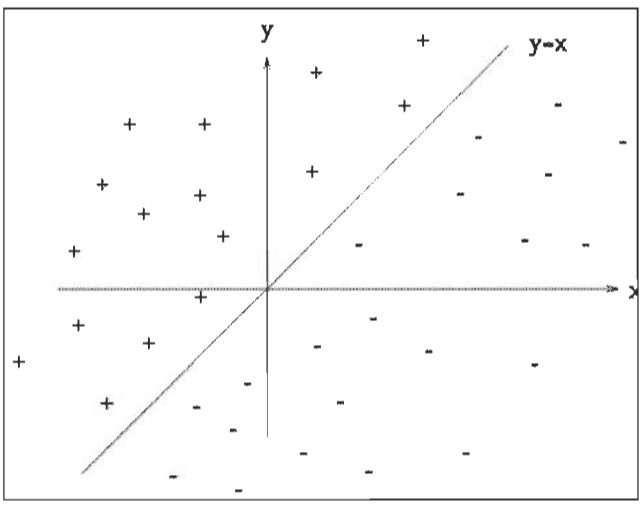


Figure 2.1 :Problème de classification à deux classes avec une séparatrice linéaire[10]

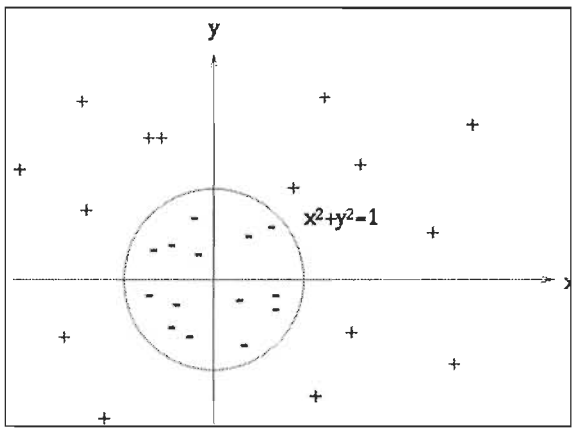


Figure 2.2 : Problème de classification à deux classes avec une séparatrice non linéaire [10]

Pour deux classes et des données linéairement séparable (figure 2.3), il y a beaucoup de séparateurs linéaires possibles. Les SVM choisissent seulement celui qui est optimal, c'est-à-dire la recherche d'une surface de décision qui soit éloignée au maximum de tout point de données. Cette distance de la surface de décision au point de données le plus proche détermine la marge maximale du classifieur. En effet, pour obtenir un hyperplan optimal, il faut maximiser la marge entre les données et l'hyperplan. [3, 10].

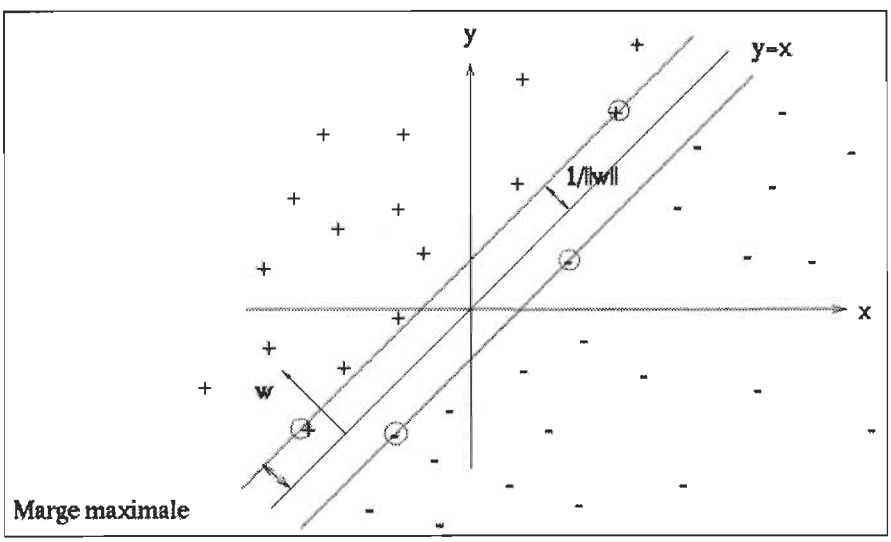


Figure 2.3 : Hyperplan optimal avec une marge maximale [10]

Pour résoudre le problème de la non linéarité séparatrice, l'idée des SVM est d'augmenter la dimension d'espace de données. Dans ce cas, il est alors probable qu'il existe un séparateur linéaire. En effet, la chance de trouver un hyperplan séparateur augmente proportionnellement avec la dimension d'espace de données. [3, 10]

Ce redimensionnement d'espace est basé sur l'utilisation de la fonction Kernel (noyau). On trouve plusieurs types de fonction noyau comme Gaussien, polynomiale et sigmoïde.[3]

2.4.2 Avantages des SVM :

Les SVM représentent plusieurs avantages, notamment ceux-ci :

1. Elles ont une base théorique solide.[11]

2. Les SVM sont efficaces dans les espaces de grande dimension.[3]

3. Différentes fonctions noyau peuvent être spécifiées. [3]

2.4.3 Inconvénients des SVM

Malgré leurs performances, les SVM représentent aussi des faiblesses, notamment celles-ci :

1. Elles utilisent des fonctions mathématiques complexes pour la classification. [3]

2. Les machines à support de vecteurs demandent un temps énorme durant les phases de test. [11]

2.5 Les réseaux de neurones :

Les réseaux de neurones ont été développés comme un modèle mathématique générique afin de modéliser les neurones biologiques. Ils comportent un certain nombre d'éléments de traitement d'information appelés neurones.[12]

Chaque neurone a son propre état interne interprété par la fonction d'activation. Il envoie son activation aux autres neurones sous forme de signaux. La connexion entre les neurones est réalisée via des liens orientés et pondérés.[12]

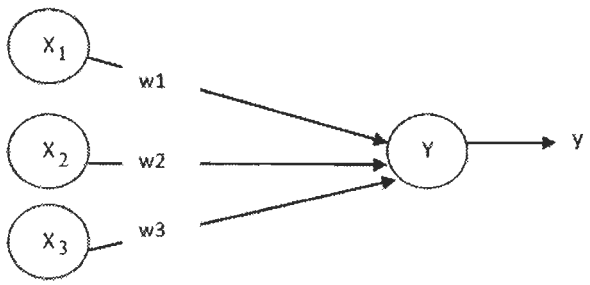


Figure 2.4 : Neurone artificiel avec une seule sortie

Le neurone Y reçoit les entrées de X_1, X_2 *et* X_3 qui ont comme valeurs de sortie x_1, x_2 *et* x_3. Les poids des liens de connexion de X_1, X_2 *et* X_3 sont w_1, w_2 *et* w_3. La valeur d'entrée de neurone Y est : $\boldsymbol{y} = w_1x_1 + w_2x_2 + w_3x_3$. Le signal de sortie y est déterminée par la fonction d'activation $f(y)$.

Les réseaux de neurones sont caractérisés par l'architecture (l'organisation des neurones), l'apprentissage (méthode de détermination des poids de connexions), et par leur fonction d'activation.[12]

2.5.1 Architecture :

Les réseaux de neurones sont souvent classifiés en deux architectures : les réseaux de neurone avec un seul niveau et multi-niveaux. Le nombre de niveaux est calculé sans prendre en considération les unités.[12]

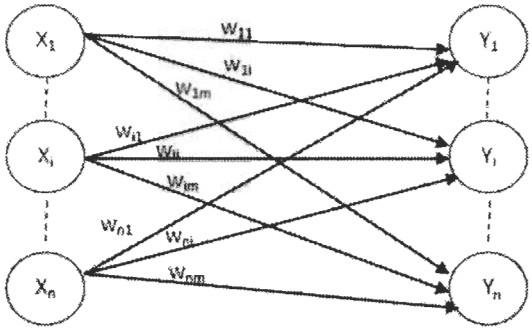


Figure 2.5 : Le réseau de neurones à un seul niveau

Les neurones de la couche d'entrée doivent uniquement passer et distribuer les entrées et ne pas effectuer de calcul. Ainsi, la seule vraie couche de neurones est celle de droite. Chacune des entrées $X_1, X_2 \ldots X_n$ est connectée à chaque neurone de la couche de sortie à travers le poids de lien. Comme chaque valeur des sorties $Y_1, Y_2 \ldots Y_n$ est calculée à partir du même ensemble de valeurs d'entrée, chaque sortie est modifiée en fonction des poids de liens.[12, 13]

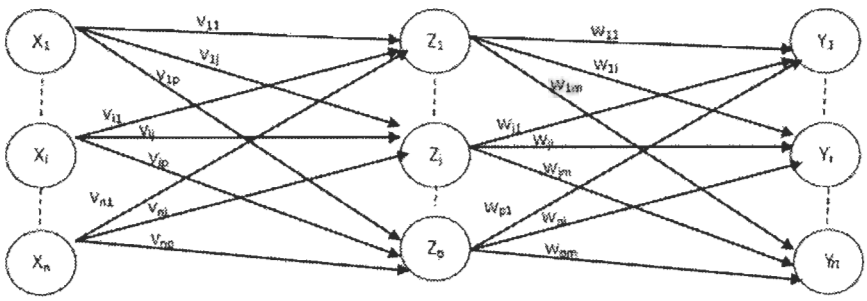


Figure 2.6 : Le réseau de neurones multi-niveaux

La figure 2.6 montre le réseau de neurones multi-niveaux ce dernier se distingue du réseau d'un seul niveau en ayant une ou plusieurs couches masquées. Dans cette structure, les nœuds d'entrée transmettent les informations aux unités dans la première couche masquée, puis les sorties de la première couche masquée sont passées à la couche suivante, et ainsi de suite. [12, 13]

n

Le réseau multi-niveaux peut également être considéré comme une cascade de groupes de réseaux d'un seul niveau. Le niveau de complexité se traduit par le nombre de réseaux monocouche qui sont combinés dans ce type de réseau. Le concepteur d'un réseau de neurones devrait considérer combien de couches cachées sont requises, selon la complexité du calcul souhaité. [12, 13]

2.5.2 L'apprentissage :

La méthode de paramétrage des poids (apprentissage) est une caractéristique importante pour distinguer différents types de réseaux de neurones. Deux modes d'apprentissage existent : l'apprentissage supervisé, et l'apprentissage non supervisé.[12]

2.5.2.1 L'apprentissage supervisé :

Dans ce type d'apprentissage, les entrées et les sorties sont fournies au préalable. Ensuite, le réseau traite les entrées et compare ses résultats aux sorties souhaitées. Les poids sont ensuite ajustés grâce aux erreurs propagées à travers le système. Ce processus se produit à plusieurs reprises tant que les poids sont continuellement améliorés. L'ensemble de données qui permet l'apprentissage est appelé l'ensemble d'apprentissage. [12, 13]

2.5.2.2 L'apprentissage non supervisé :

Dans l'apprentissage non supervisé, le réseau est fourni avec des entrées mais pas avec les sorties souhaitées. Le système lui-même doit alors décider quelles fonctionnalités il utilisera pour regrouper les données d'entrée. C'est ce qu'on appelle souvent l'auto-organisation ou l'adaptation.[12, 13]

2.5.3 Fonction d'activation :

La fonction basique d'un neurone artificiel est d'effectuer une somme de toutes les données d'entrées afin de produire une fonction de sortie. Cette fonction est la fonction d'identité.[12]

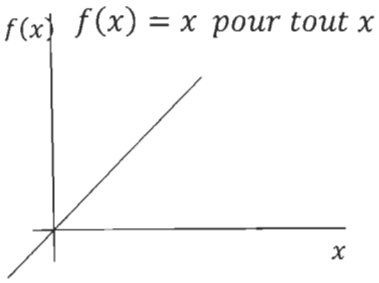


Figure 2.7 : Fonction d'identité

Une autre fonction d'activation est la fonction d'un seuil . La valeur de sortie est 1 lorsque la somme pondérée des valeurs d'entrée est supérieure ou égale à θ, sinon la valeur est 0. [12]

$$f(x) = \begin{cases} 1 \ si \ x \geq \theta \\ 0 \ si \ x < \theta \end{cases}$$

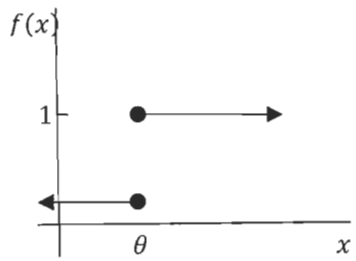


Figure 2.9 : Fonction d'un seuil θ

La fonction sigmoïde de 0 à 1 est souvent utilisée comme fonction d'activation pour les réseaux de neurones dans lesquels les valeurs de sortie désirées sont soit binaires soit dans un intervalle compris entre 0 et 1. [12]

$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$

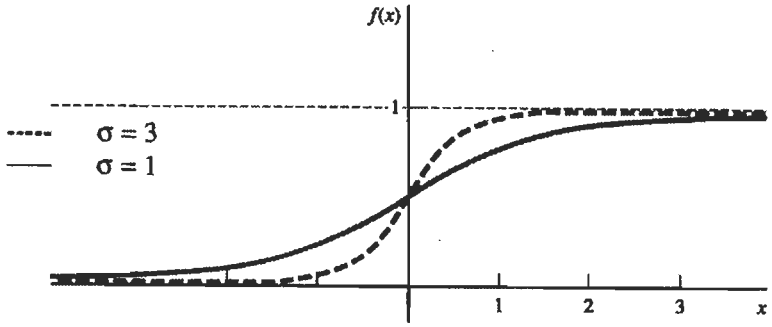


Figure 2.9 : Fonction sigmoïde

2.5.4 Topologie des réseaux de neurones :

Nous distinguons deux types de topologies de réseaux de neurones :

2.5.4.1 Propagation vers l'avant de l'information (Feed-forward) :

Le flux de données entre les unités d'entrée et de sortie est strictement alimenté vers l'avant. Le traitement des données peut s'étendre sur plusieurs unités, mais il n'existe pas de connexions bouclées.[12, 14]

2.5.4.2 Récurent (Feed-back connections) :

Contrairement à la propagation vers l'avant de l'information (Feed-forward), les propriétés dynamiques de réseaux de neurones sont importantes. Dans certains cas, les valeurs d'activation des unités subissent un processus de relaxation tel que le réseau évolue vers un état stable dans lequel ces activations ne changent plus. Dans d'autres applications, le changement des valeurs d'activation des neurones de sortie est significatif, de sorte que le comportement dynamique constitue la sortie du réseau. [12, 14]

2.5.5 Les réseaux de Perceptron :

La forme la plus simple d'un réseau de neurone est le perceptron. Ce réseau est considéré parmi les premiers réseaux de neurones. Il a été inventé en 1957 par Rosenblatt. Le perceptron se compose d'un neurone artificiel à poids ajustable et d'un seuil. Il n'a qu'une seule sortie à laquelle toutes les entrées sont connectées. Les entrées et la sortie sont booléennes.[3, 15, 16]

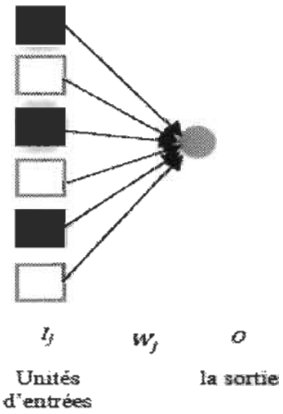


Figure 2.10 : réseaux de perceptron [17]

Dans ce type de réseaux, seulement les poids entre les unités d'entrées et la sortie peuvent être modifiés, tandis que la sortie de neurone ne peut prendre que deux états : -1 et 1 ou 0 et 1.[3]

2.5.6 Les réseaux Hopfield :

Développé par Hopfield en 1982, ce type de réseau a une topologie récurrente à temps discret dont un seul neurone est mis à jour à chaque unité de temps. Ce réseau est constitué de N neurones qui prennent deux états -1 et 1 ou 0 et 1, tous interconnectés. L'entrée totale d'un neurone i est donc : $I_i = \sum_j w_{ij} V_j$. [3, 18]

Où :

w_{ij} , est le poids de la connexion du neurone i à .

V_j, est l'état du neurone j.

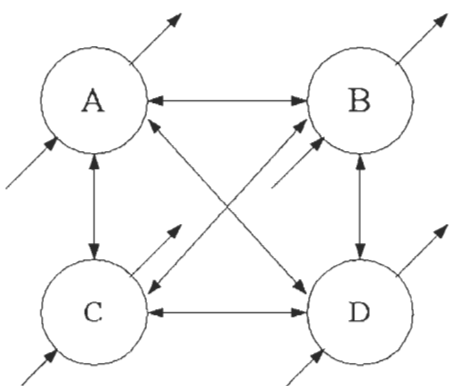


Figure 2.11 : réseaux Hopfield **[18]**

Le réseau utilise la règle Hebb (Hebb, 1949) comme loi d'apprentissage. Cette loi se base sur le principe suivant : une synapse améliore son activité si et seulement si l'activité de ces deux neurones est corrélée.[3, 18]

2.5.7 Les réseaux du perceptron multicouche (multilayer perceptron MLP) :

Les perceptrons multicouches (MLPs) également appelés les réseaux profonds (Feed-forward) effectuent la propagation vers l'avant de l'information. Chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système.[19, 20]

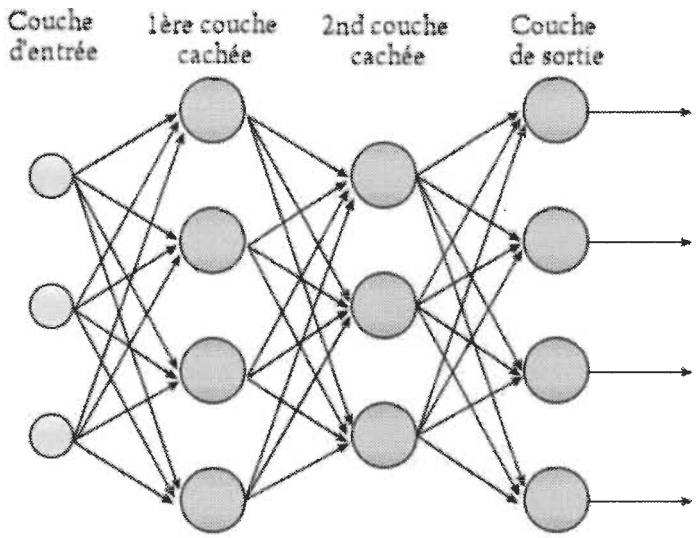


Figure 2.12 : réseaux du perceptron multicouche [20]

Les MPLs utilisent un algorithme de rétropropagation du gradient, l'objectif étant de minimiser l'erreur quadratique. La modification des poids est propagée de la couche de sortie jusqu'à la couche d'entrée.[3]

2.5.8 Avantages des réseaux de neurones :

Les réseaux de neurones représentent des avantages, tels que :

1. Les réseaux de neurones sont souples et génériques. Ils peuvent résoudre différents types de problèmes dont le résultat peut être : une classification, analyse de données,etc.[21]

2. Ils traitent des problèmes non structurés sur lesquels aucune information n'est disponible à l'avance.[21]

3. Les réseaux neuronaux se comportent bien parce que même dans des domaines très complexes, ils fonctionnent mieux que les arbres de statistique ou de décision.[22]

4. Les réseaux de neurones fonctionnent sur des données incomplètes ou bruitées. Cette lacune d'information peut être complétée par l'ajout d'autres neurones à la couche cachée.[21]

2.5.9 Inconvénients des réseaux de neurones :

Les réseaux de neurones ont aussi des inconvénients, tels que :

1. La lenteur d'apprentissage.[3]

2. La difficulté de choisir des valeurs initiales des poids de connexion ainsi que l'adaptation du pas d'apprentissage.[22]

3. L'apprentissage au détriment de la généralisation.[22]

4. En cas d'erreur dans les résultats de sorties, l'utilisateur n'a aucune information sur le fonctionnement interne.[3]

2.6 Les cartes auto organisatrices de Kohonen (SOM) :

Développées par T. Kohonen en 1982, les cartes auto organisatrice (SOM) sont des types de réseaux de neurones. Elles constituent un outil efficace et performant qui permet de classifier des échantillons par rapport à leurs similarités.[3, 10]

Elles fournissent un moyen pour représenter des données multidimensionnelles dans des espaces de dimensions faibles (habituellement une ou deux dimensions). En outre, elles créent un réseau qui stocke l'information.[23]

2.6.1 L'architecture des cartes auto organisatrices de Kohonen :

La figure 2.4 montre un réseau Kohonen de 6 x 6 nœuds dont chacun d'eux est entièrement connecté à la couche d'entrée. Pour cet exemple, on compte 6 x 6 x 6 = 216 connexions. Les nœuds sont organisés de cette manière, comme une grille bidimensionnelle, qui permet de visualiser facilement les résultats. Il n'y a pas de connexions latérales entre les nœuds dans le réseau. Dans cette architecture, chaque nœud de la carte a une coordonnée unique (i, j). Cela facilite la référence d'un nœud dans le réseau et le calcul des distances entre les nœuds.[23, 24]

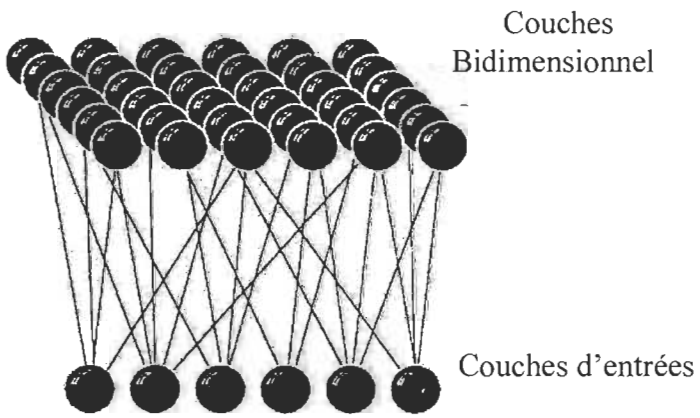


Figure 2.13 : L'architecture des cartes auto-organisatrices de Kohonen [25]

2.6.2 L'algorithme des cartes auto-organisatrices de Kohonen :

Le déroulement de l'algorithme de Kohonen est résumé dans les étapes suivantes : [3]

1. Initialiser les neurones en fonction de la topologie choisie.
2. **Pour chaque** vecteur d'entré x
3. Lire le vecteur x.
4. **Pour chaque** vecteur de neurone w_k
5. Calculer la distance euclidienne : $D_k = \|x - w_k\|$.
6. Le gagnant $k = argMin\ (D_k = \|x - w_k\|\ )$ // La plus petite distance.
7. Tirer vers x le gagnant k et les autres neurones du voisinage, centre sur k

Algorithme 2.3 : L'algorithme des cartes auto-organisatrices de Kohonen.

2.6.3 Avantages des cartes auto-organisatrices de Kohonen :

Les cartes auto-organisatrices de Kohonen comportent des avantages tels que :

1. L'algorithme présente des opérations simples.[3]

2. L'algorithme s'avère très léger en termes de coût de calculs.[26]

3. Permet une visualisation graphique des résultats. [3]

2.6.4 Inconvénients des cartes auto-organisatrices de Kohonen :

Les des cartes auto-organisatrices de Kohonen comportent des inconvénients tels que :

1. Le voisinage dans les cartes auto-organisatrices est fixe.[26]

2. Une liaison entre neurones ne peut être cassée même pour mieux représenter des données discontinues. [26]

3. 1. Leur temps énorme de convergence.[3]

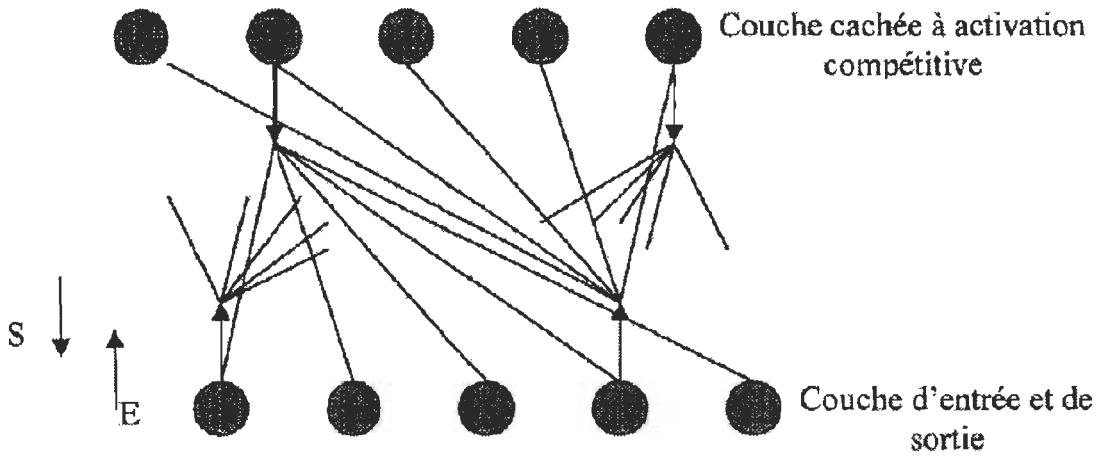
2.7 Le réseau à architecture évolutive ART :

ART ((Adaptive Resonance Theory) est une théorie développée par Grossberg et Carpenter. Il décrit un certain nombre de modèles de réseaux neuronaux qui utilisent des

méthodes d'apprentissage supervisées et non supervisées. Il existe plusieurs versions de réseaux (ART1, ART2, ART3). Le réseau ART1 est un réseau à entrées binaires.

2.7.1 Architecture :

Le réseau de neurone ART1 a une architecture multi-niveaux (figure 2.14) : la couche d'entrée (unités d'entrées), la couche cachée (ensemble des unités cachées invisible aux utilisateurs) et la couche de sortie (unités de sortie). Dans ce type de réseau, la couche d'entrée et la couche de sortie se sont superposées l'une sur l'autre pour former une seule couche baptisé entrée-sortie. Les neurones de la couche entrée-sortie sont tous connectés aux neurones de la couche cachée ; réciproquement, chaque neurone de la couche cachée à son tour est connecté avec tous les neurones de la couche entrée-sortie. La couche cachée est une couche compétitive dans laquelle tous les neurones sont reliés les uns aux autres. Typiquement, les interconnexions entre les neurones dans la couche compétitive ne sont pas représentées dans les diagrammes



d'architecture de ces réseaux.[12, 27]

Figure 2.14 : Architecture de réseau ART1

2.7.2 Apprentissage :

Le réseau ART1 procède sur des données binaires. Les données sont présentées en série devant la couche d'entrée. La figure 2.15 montre une donnée E soumise au réseau. Ensuite, et grâce aux liens de connexion de la couche entrée-sortie vers la couche compétitive, la donnée E se propage dans cette dernière. Les neurones de cette couche entrent en compétition, d'où le nom de la couche de compétition, et à la fin de cette compétition, un seul neurone gagne soit j dans ce cas. Ce gagnant est considéré par le réseau comme le plus représentatif du vecteur d'entrée E. Après avoir choisi le neurone j et grâce, cette fois-ci grâce aux liens de connexions de la couche compétitive vers la couche entrée-sortie, un vecteur S binaire est généré par le neurone j sur la couche entrée-sortie. Dès la présence de vecteur S sur la couche d'entrée-sortie, un test de similarité est effectué entre E et S : si $\frac{\|E\|}{\|S\|} < \rho$ (ρ : seuil de similarité). $\|E\|$ représente la norme de vecteur E par exemple $\|1,0,1,1\| = 3$. Le neurone gagnant est choisi en tant que la classe de vecteur d'entrée E, dans ce cas, la modification des poids des connexions du neurone gagnant a pour effet de consolider ses liens d'activation avec l'entrée E, sinon ($\frac{\|E\|}{\|S\|} \geq \rho$) le procédé reprend sans le neurone j (le gagnant de l'étape précédente). Si tous les neurones de la couche compétitive ne correspondent pas à l'entrée E, un nouveau neurone caché est ajouté qui est initialisé comme représentant de la classe du vecteur d'entrée E.[12, 27]

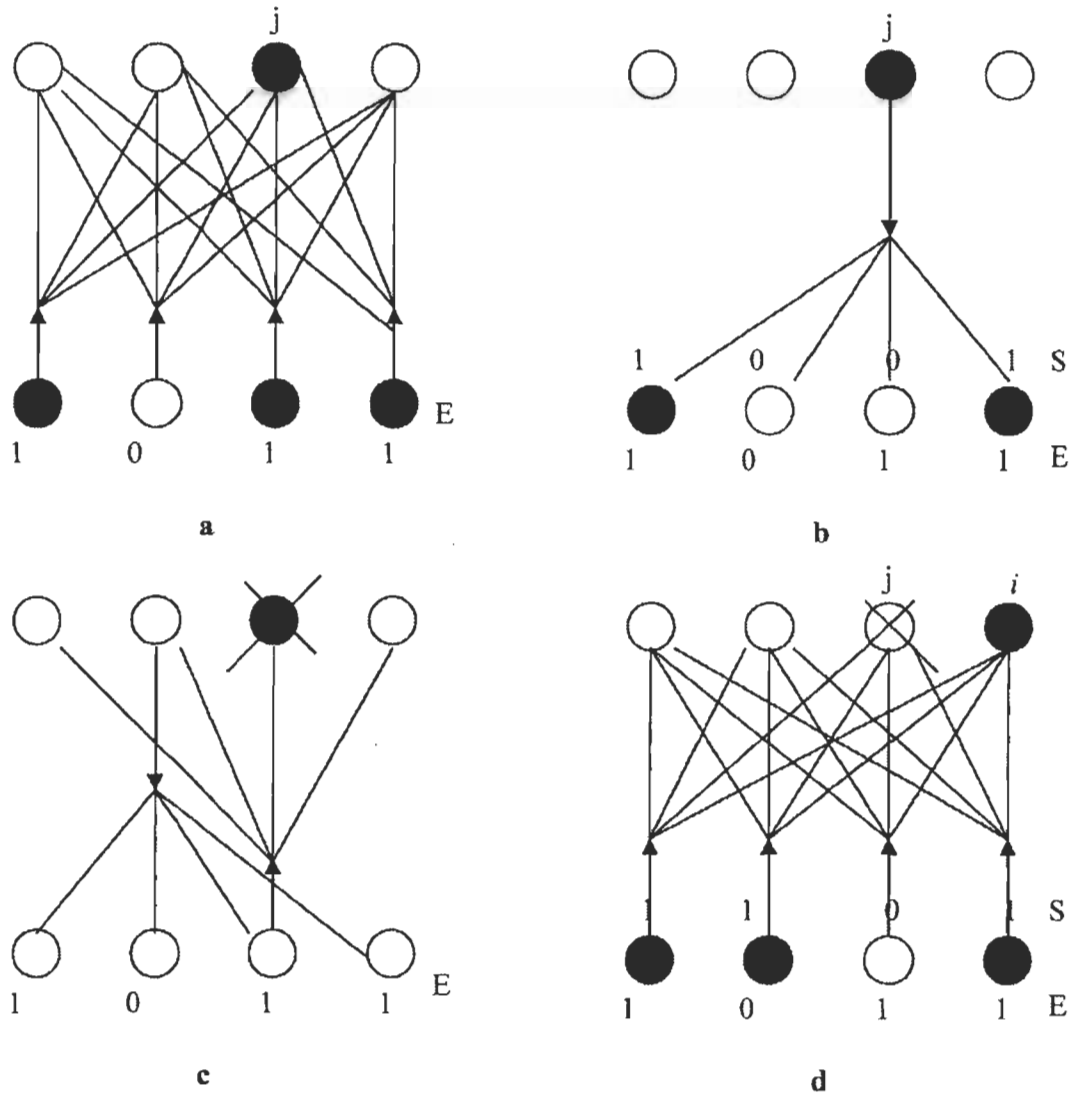


Figure 2.15 : Apprentissage du réseau ART1

2.7.2.1 Algorithme :

1. **Input :** $0 < \rho \leq 1$ (paramètre de similarité), $L \geq 0$ (paramètre utilisé dans le calcul des poids) et A (une base d'apprentissage).

2. **Pour** chaque vecteur E de l'ensemble A **Faire**

3. Calcul du neurone gagnant S_j.

4. **Si** $\frac{\|E\|}{\|S_j\|} < \rho$ **Alors**

5. L'unification est réalisée. Aller à l'étape 15.

6. **Sinon**

7. Le neurone gagnant N_j est inhibé (c'est-à-dire qu'il devient inaccessible à la prochaine étape).

8. **Si** neurones non inhibés **Alors**

9. Retour à l'étape 15.

10. **Sinon**

11. Un nouveau neurone caché est créé, initialisé comme représentant de la classe.

12. Aller à l'étape 6.

13. **Finsi**

14. **Finsi**

15. Modification des poids : $b_{ij}(nouveau) = \frac{L}{L-1+\|E\|}$.

16. **Fin pour**

Algorithme 2.4 : L'algorithme d'apprentissage de ART 1. **[12]**

La valeur du seuil ρ contrôle le degré d'unification (similarité) recherché entre les formes à classer et les prototypes des classes. Plus la valeur du seuil est grande, meilleure est l'adéquation recherchée.[12]

2.7.3 Avantages des ARTs :

Les ARTs représentent plusieurs avantages tels que :

1. Le paramétrage des ARTs est facile où ils ne requièrent qu'un seuil de vigilance. [12]

2. Les ARTs peuvent effectuer un apprentissage en temps réel. [12, 27]

3. Ils peuvent fonctionner dans des environnements dynamiques. [12]

2.7.4 Inconvénients des ARTs :

Les ARTs ont aussi des inconvénients tels que :

1. Le temps de réponse élevé grâce à une approche à deux couches. [12]

2. Le même seuil de vigilance est appliqué à tous les clusters. [12, 27]

3. Les ARTs conduit souvent à la dégradation. Cette dernière est définie par la capacité d'un ART à fournir une classification utile en présence de défauts. [12, 27]

2.8 Les algorithmes génétiques :

Les algorithmes génétiques ont été initialement développés par Holand (1975). Les AGs utilisent un vocabulaire similaire à celui de la génétique naturelle. Ainsi, une population est un ensemble d'individu et ceci sera résumé, bien souvent, par un seul chromosome. Les chromosomes sont constitués de gènes qui contiennent les caractères de l'individu. On trouvera aussi les principes de sélection, de croisement, de mutation, etc.[28]

Chaque point dans l'espace d'état se traduit par un chromosome en associant une valeur du critère à optimiser. Ensuite, on applique l'algorithme génétique pour sélectionner les meilleurs individus à partir d'une population d'individus générée aléatoirement, tout en assurant une exploration efficace de l'espace d'état.[28]

2.8.1 Terminologie [29, 30]:

- **Gène** : C'est la suite de symboles qui codent la valeur d'une variable. Il correspond souvent à un seul symbole.
- **Chromosome** : Une séquence finie de gènes, qui sont pris dans un alphabet et qui répondent au problème étudié.
- **Individu** : Une des solutions potentielles bien souvent représentée par un seul chromosome.

- **Fitness d'un individu** : Une valeur associée à l'individu pour mesurer sa qualité dans un problème bien déterminé.
- **Population** : L'ensemble des solutions potentielles qu'utilisent les AGs.
- **Espace de recherche** : Définit l'ensemble des configurations possibles des paramètres de la fonction à optimiser.

2.8.2 Principes des algorithmes génétiques :

Généralement, un algorithme génétique a cinq parties de bases selon Michalewicz [31] :

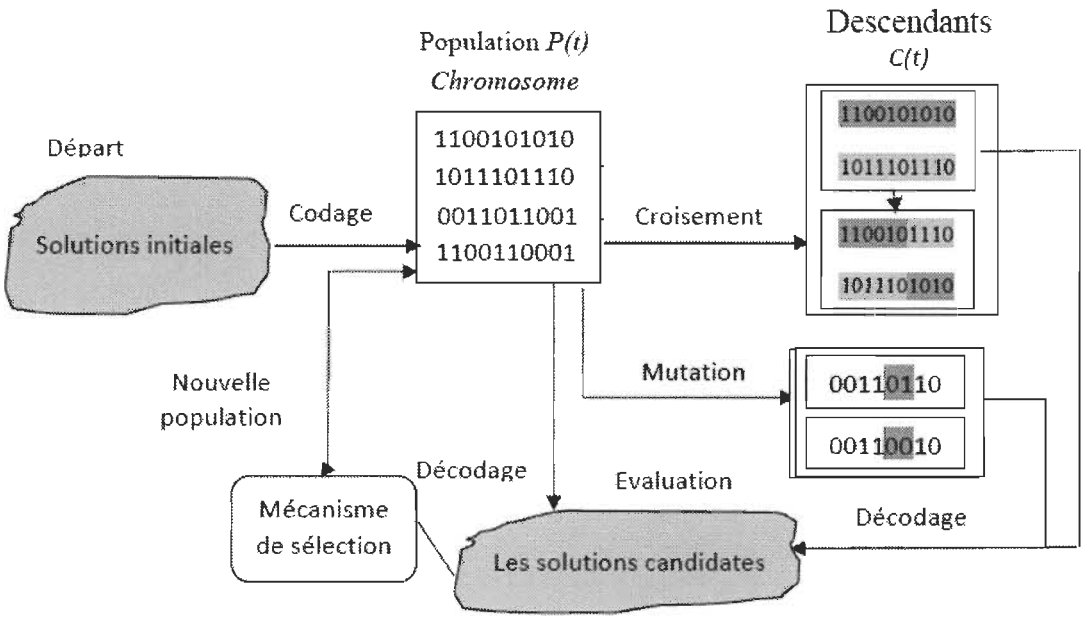


Figure 2.16 : Structure générale d'un algorithme génétique [31]

La figure 1 montre la structure générale d'un algorithme génétique. Nous pouvons constater les étapes suivantes [31] :

1. Au début, on commence par une population initiale qui représente les solutions d'un problème donné.

2. On applique l'opération de codage sur chaque solution initiale afin de présenter les individus sous forme des chromosomes qui représentent des solutions. A la fin de cette opération on obtient une population $P(t)$.

3. On applique aléatoirement l'opération de croisement avec une probabilité P_c, sur deux chromosomes de la population $P(t)$, et par la suite deux nouveaux chromosomes ont été créés. Idem pour la mutation avec une probabilité P_m on obtient deux nouveaux chromosomes.

4. L'application de l'opération de décodage à chaque chromosome de population $P(t)$ ainsi qu'aux nouveaux chromosomes, attribue un fitness à chacun,

5. Le mécanisme de sélection choisit les chromosomes qui seront insérés dans la prochaine génération en fonction de leurs fitness.

6. Si la condition d'arrêt est fausse, on répète à partir de la deuxième étape, mais avec la population $P(t+1)$, sinon l'algorithme s'arrêteras et la meilleur solution s'affichera.

2.8.3 Pseudo code d'un Algorithme génétique :

Le pseudo code de l'algorithme génétique est donné dans l'algorithme 2.4 :

Entrée : données du problème, les paramètres de GA

Sortie : la meilleure solution

Début

t ←0;

Initialisation de la population P(t) ;

Évaluation de la population P(t) ;

Tant que (la condition d'arrêt non satisfaite) **faire**

Créer C(t) à partir de P(t) par croisement ;

Créer C(t) à partir de P(t) par mutation ;

Évaluer C(t) par décodage ;

Sélectionner P(t+1) à partir P(t) et C(t) avec le mécanisme de sélection ;

t ←t+1;

Fin tant que

Fin

Algorithme 2.5 : Pseudo code d'un Algorithme génétique **[31]**

Les Algorithmes génétiques se basent principalement sur trois opérations pour permettre la reproduction des chromosomes [29, 30]:

1. **Sélection** :

La sélection est une méthode qui sélectionne aléatoirement des chromosomes à partir de la population en fonction des valeurs de la fonction d'adaptation. Elle permet de donner aux meilleurs individus une chance de contribuer à la génération suivante.

2. **Croisement** :

Le croisement est le processus de prendre deux parents et de produire à partir d'eux des enfants. Il s'agit d'un processus essentiel pour explorer l'espace des solutions possibles. Dans la littérature il existe plusieurs opérateurs de croisement. Ils diffèrent selon le type de codage et la nature du problème.

3. **Mutation :**

C'est un processus qui sert à appliquer un changement mineur de code génétique a un individu pour assurer la diversité et ainsi éviter de tomber dans des optimums locaux (ce sont des solutions optimales au sein d'un ensemble voisin de solutions candidates). Dans un cas général, la mutation en codage binaire est la modification aléatoire avec une faible probabilité, de la valeur d'un caractère de la chaîne.

2.8.4 Avantages des algorithmes génétiques :

Les algorithmes génétiques représentent plusieurs avantages tels que :

1. Leurs convergences ne dépendent pas de la valeur initiale[29].

2. Ils permettent de déterminer l'optimum global de la fonction objectif [29].

3. Ils représentent des méthodes génériques qui peuvent optimiser une large gamme de problèmes différents [30].

4. Leur capacité à faire plusieurs calculs en parallèle [3].

2.8.5 Inconvénients des algorithmes génétiques :

Les algorithmes génétiques ont aussi des inconvénients tels que :

1. Le temps de calcul est énorme puisqu'ils manipulent plusieurs solutions en parallèle[3].

2. La recherche pour la solution optimale se limite généralement autour d'un minimum qui n'est pas forcement l'optimum attendu. On parle dans ce cas de convergence prématurée [30].

3. L'efficacité d'un algorithme génétique dépend beaucoup de la méthode de croisement et du type de codage choisis [29].

2.9 Apprentissage profond (Deep learning) :

Depuis 2006, l'apprentissage profond est apparu comme une nouvelle zone de recherche de l'apprentissage automatique. Au cours des dernières années, les techniques développées dans l'apprentissage profond ont déjà eu un impact sur les travaux de traitement des signaux et de l'information, y compris les aspects de l'apprentissage automatique et l'intelligence artificielle.[19]

L'apprentissage profond est une classe de techniques d'apprentissage automatique qui modélisent avec un haut niveau d'abstraction des données grâce à des

architectures multiples niveaux. Les caractéristiques et les concepts de niveau supérieur sont donc définis en termes de niveaux inférieurs, et une telle hiérarchie est appelée architecture profonde.[32, 33]

Afin de bien comprendre l'apprentissage profond, il faut avoir une solide compréhension des principes de base de l'apprentissage automatique.

2.9.1 Apprentissage automatique :

Chaque algorithme qui est capable d'apprendre de données est un algorithme d'apprentissage automatique. Ce dernier est dit capable d'apprendre de données si sa performance aux tâches dans T, mesurée par la performance P, s'améliore avec l'expérience E.[19, 32, 33]

A) La tâche, T : [19, 32, 33]

De nombreux types de tâches peuvent être résolus avec l'apprentissage automatique à titre d'exemple : la classification, la régression, et la traduction, etc. Dans la classification, l'algorithme spécifie à laquelle des catégories k certaines entrées appartiennent (la reconnaissance d'objet est un exemple de classification, où l'entrée est une image et la sortie est un code numérique identifiant l'objet dans l'image). La régression, prédit une valeur numérique étant donnée une entrée (la prévision du montant réclamé par une personne assurée). La traduction est une tâche qui consiste à convertir une séquence de symboles écrite dans une certaine langue à une autre langue.

B) La mesure de performance, P : [19, 32, 33]

Nous devons concevoir une mesure quantitative de performance pour évaluer les capacités d'un algorithme d'apprentissage automatique. Par exemple, pour la tâche classification, nous mesurons souvent la précision du modèle (une proportion d'exemples pour lesquels le modèle produit la sortie correcte). Il est souvent difficile de choisir une mesure de performance qui corresponde bien au système.

C) L'expérience, E :[19, 32, 33]

Les algorithmes d'apprentissage automatique peuvent être classés en deux catégories : non supervisés ou supervisés. Les algorithmes d'apprentissage supervisés expérimentent une base de données d'apprentissage contenant des exemples prélassés. Tandis que, les algorithmes d'apprentissage supervisés expérimentent un ensemble de données pour apprennent la structure de ces données sans utilisation d'une base de données d'apprentissage.

Exemple 1 : la classification

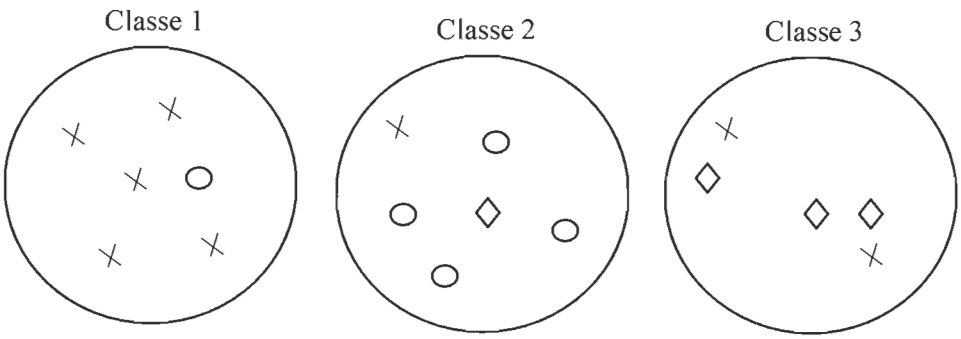


Figure 2.17 : résultat de classification.

La figure 2.17 montre un résultat de classification sur trois types de données à savoir : x (croix), o (cercle), ◊ (losange). Le classifieur choisi est non supervisé c'est-à-dire il n'y pas besoin d'une base d'apprentissage.

Afin d'évaluer la performance de classifieur une mesure est choisi en l'occurrence : la pureté qui est une mesure simple et transparente. La pureté permet de mesurer la qualité externe de classification.[34]

Pour calculer la pureté on compte le nombre de données correctement attribués à une classe divisé par N (nombre total de données). Dans cet exemple, cinq (05) croix sont correctement attribuées à la classe 1, quatre (04) cercles sont correctement

attribuées à la classe 2 et trois (03) losanges sont correctement attribués à la classe 3. Ainsi, la pureté est $\frac{1}{N}\ x\ (5+4+3)\ \approx 0{,}71$, tel que $N = 17$.

Exemple 2 : les réseaux de neurones profonds (Deep Neural Network, DNN)

Les DNNs (Deep Neural Network) effectuent la propagation vers l'avant de l'information (Feed-forward). Ils comprennent une couche d'entrée, plusieurs couches cachées et la couche de sortie (figure 2.18). Les couches cachées et la couche de sortie se composent de nœuds où la sortie d'une couche est une entrée de la couche suivante. Les nœuds effectuent une fonction d'activation linéaire suivie par une fonction d'activation non linéaire sur la valeur d'entrée (figure 2.19). Dans la figure 2.19, la valeur d'entrée de nœud $y^{(l)} = \sigma(W^{(l)}.y^{(l-1)} + b^{(l)})$ où $W^{(l)}.y^{(l-1)} + b^{(l)}$ est une fonction d'activation linéaire et $\sigma(W^{(l)}.y^{(l-1)} + b^{(l)})$ une fonction d'activation non linéaire. [32]

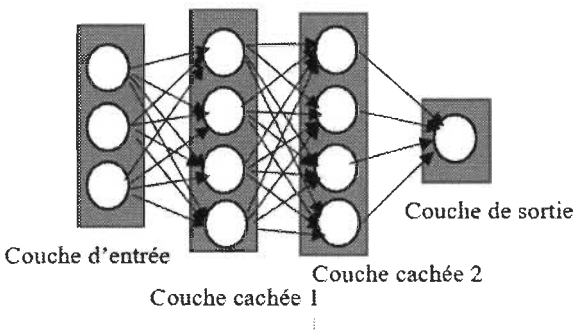


Figure 2.18 : l'architecture des DNNs

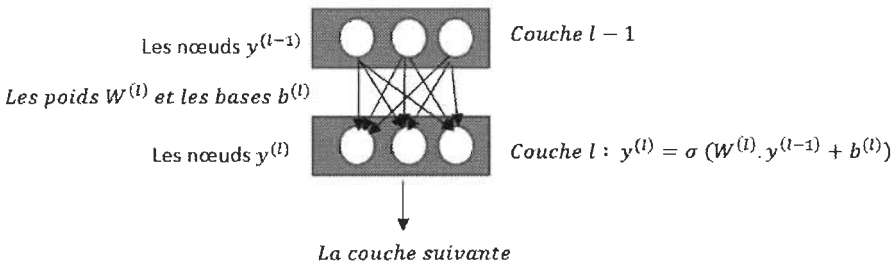


Figure 2.19 : la relation entre une couche inférieure et une couche supérieure.

L'apprentissage des DNNs par la rétropropagation du gradient (le gradient de l'erreur pour chaque neurone est calculé de la dernière couche vers la première) entraînant des problèmes fondamentaux. Autrement dit, une fois que les erreurs sont rétropropagées aux premières couches, elles deviennent minuscules et l'apprentissage devient inefficace. Afin de surmonter ce problème d'autres techniques d'apprentissage sont introduites (voir section 2.9.3.3) [19, 32].

2.9.3 La catégorisation de l'apprentissage profond :

Selon la façon dont les architectures et les techniques sont destinées à être utilisées, on peut classer globalement l'apprentissage profond en trois grandes catégorie[32] :

2.9.3.1 Les réseaux profonds pour l'apprentissage non supervisé :

Ils sont destinés à capturer une relation élevée des données observées pour l'analyse de motifs ou la synthèse quand aucune information sur les étiquettes de sorties n'est disponible.[32]

Exemple : les autoencoders

L'autoencoder est un type spécial du DNN (Deep Neural Networks) sans classe étiquetée, dont les vecteurs de sortie ont la même dimensionnalité que les vecteurs d'entrées. Il est souvent utilisé dans l'encodage de données.[32]

Un autoencoders a typiquement une couche d'entrée (couche L_1) qui représente les vecteurs de données ou de caractéristique, une ou plusieurs couches cachées qui représentent la caractéristique transformée (couche L_2) et une couche de sortie qui correspond à la couche d'entrée (couche L_2). Lorsque le nombre de couches cachées est supérieur à un, l'autoencoder est considéré comme profond. La dimension des couches peut être soit plus petite (lorsque l'objectif est la compression) ou grande (lorsque l'objectif est d'augmenter la dimension d'espace).[19, 32]

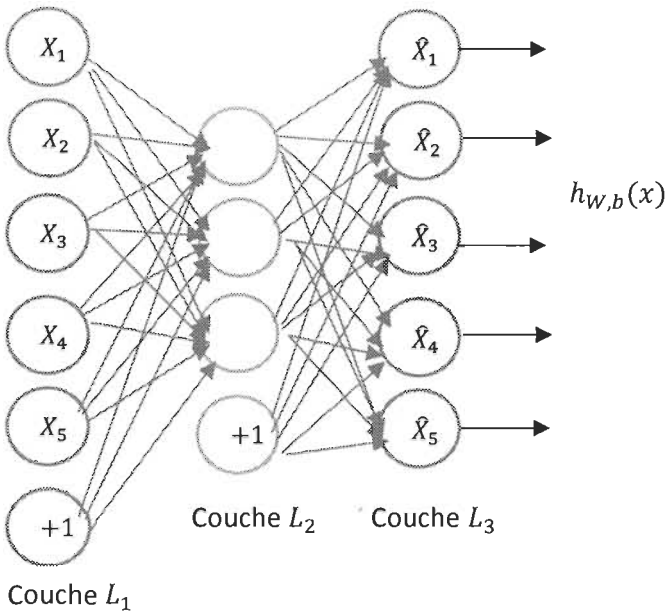


Figure 2.20 : L'architecture des Autoencoders.[35]

2.9.3.2 Les réseaux profonds pour l'apprentissage supervisé :

Ils sont destinés pour fournir directement une puissance discriminative pour la classification des motifs, souvent en caractérisant les distributions postérieures des classes conditionnées sur les données visibles.[32]

Exemple : DSN (Deep Stacking Networks)

Le DSN (Deep Stacking Networks) a été introduit avec une architecture légèrement différente de DNN (Deep Neural Networks). Il consiste réellement en petits sous-réseaux avec une seule couche cachée.[14, 32]

Chaque couleur indique un sous-réseau, également appelé *module*. La sortie de n'importe quel *module* peut être copiée sur des couches supérieures pour créer une architecture convolutée (Les lignes pointillées désignent des couches copies). La même architecture se répète dans les *modules*, une couche d'entrée linéaire suivie d'une couche

cachée non linéaire, qui est reliée à une couche de sortie linéaire. La matrice de poids de la couche inférieure, que nous désignons par W, relie la couche d'entrée et la couche cachée. La matrice de poids de la couche supérieure, que nous désignons par U, relie la couche cachée avec la couche de sortie. [14, 32]

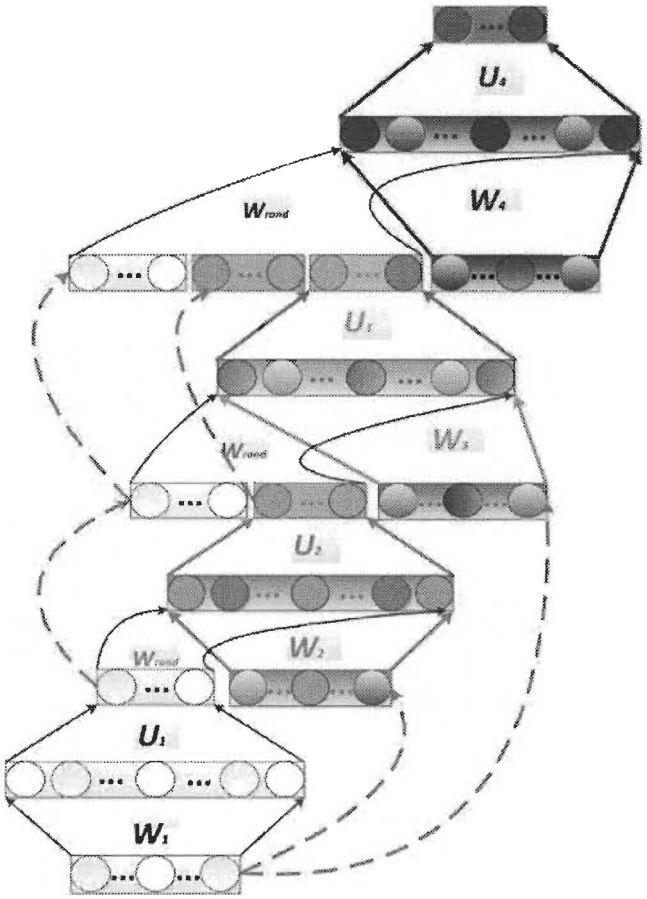


Figure 2.21 : L'architecture des réseaux DSN [14]

2.9.3.3 Les réseaux profonds hybrides :

Dans cette catégorie, l'objectif est la discrimination qui est assistée par les résultats des réseaux profonds non supervisés à titre d'exemple les autoencoders. Autrement dit, les autoencoders sont utilisés pour l'apprentissage de DNN.[32]

Exemple : SAE-DNN (Stacked autoencoder - Deep Neural Network)

Le SAE-DNN consiste à deux types de réseau de neurones profond à savoir le SAE et le DNN. Le SAE est constitué de plusieurs couches d'autoencoders (voir section 2.9.3.1) dans lesquels les sorties de chaque couche sont reliées aux entrées de la couche suivante. Ces autoencoders entrainent les couches cachées de DNN l'un après l'autre (figure 2.20). L'apprentissage de DNN comprend deux étapes (figure 2.20). À la première étape, l'« autoencoder 1» subit un apprentissage non supervisé. Ensuite, la « couche h1 » de DNN est initialisée par les poids d'« autoencoder 1» après son apprentissage. Par la suite, les poids de « couche h1» de premier autoencoder deviennent les entrées de la seconde (couche h2) et ainsi de suite. À la deuxième étape, l'apprentissage supervisé de DNN s'amorce. [14]

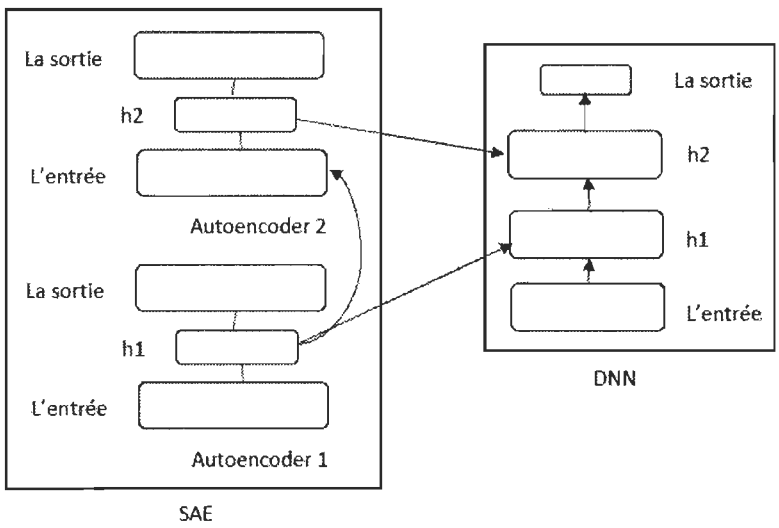


Figure 2.22 : L'architecture de SAE-DNN [14]

2.9.4 Avantages des réseaux profonds :

Les réseaux profonds représentent des avantages tels que :

1. Les réseaux profonds sont capables d'apprendre des fonctions complexes. [14]
2. Ils possèdent de bonnes capacités de généralisation. [32]

2.9.5 Inconvénients des réseaux profonds :

Les réseaux profonds ont aussi des inconvénients tels que :

1. Les réseaux profonds nécessitent une grande quantité de données. [32]

2. Ils sont extrêmement coûteux en apprentissage. [14]

2.10 Conclusion :

Dans ce chapitre, nous avons présenté une vue détaillée des principales méthodes de classification qui existent, ainsi que les avantages et les inconvénients de chacune d'entre elles.

Le chapitre suivant mettra en lumière le modèle vectoriel dans lequel nous présenterons les concepts mathématiques de ce modèle, le schéma de pondération TF-IDF, ainsi que ses avantages et ses inconvénients.

Chapitre 3

Modèle vectoriel

3.1 Introduction :

Nous présentons ici le modèle vectoriel, vu son importance dans la recherche d'information. Celui-ci, a été introduit par Gerard Salton [36]. Il sert à représenter chaque document et chaque requête par un vecteur ainsi qu'il calcule un coefficient de similarité entre ces vecteurs.

3.2 Espace de documents :

Un espace de documents est un groupe de documents, dans lequel chaque document est représenté par un vecteur de termes sous forme de :

$$D = (t_0; w_{d_0}, t_1; w_{d_1}; \dots, t_t; w_{d_t})$$

Chaque t_k représente un terme attribué au document D et w_{d_k} le poids de terme t_k. Ces termes peuvent être pondérés en fonction de leur importance, ou non pondérées avec des poids limités à 0 et 1 [37] [36].

Afin de mieux répondre aux requêtes d'utilisateur (expressions booléennes), les moteurs de recherche, souvent, les considèrent comme des documents. Ainsi, les requêtes seraient représentées sous forme de vecteur :

$$Q = (q_0; w_{q_0}, q_1; w_{q_1}; \dots, q_t; w_{q_t})$$

Chaque q_k représente un terme attribué au requête Q et w_{q_k} le poids de terme q_k dans le document Q [36].

Exemple :

La figure 3.1 montre un exemple d'un espace de documents. Chaque document, ou requête, est représenté par un vecteur caractérisé par les poids des termes T_1, T_2 et T_3 contenus dans les documents D_1 , D_2 , D_3

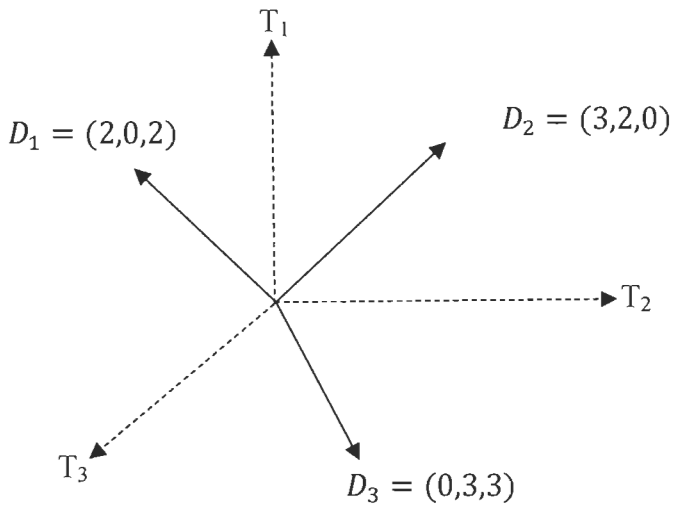


Figure 3.1 : Espace de documents de trois dimensions

Le tableau 3.1 illustre une représentation matricielle d'espace de documents.

	T_1	T_2	T_3
D_1	0	3	3
D_2	3	2	0
D_3	2	0	2

Tableau 3.1 : représentation matricielle

3.3 Coefficient de similarité :

Le coefficient de similarité, communément appelé similarité cosinus, permet de calculer la similarité entre le document et la requête en déterminant le cosinus de l'angle entre eux. Cette métrique est souvent utilisée dans la recherche d'information [36].

Étant donné deux vecteurs de représentations des équations (1) et (2), La similarité cosinus requête-document peut être calcule par cette formule .[36]

$$similarite\ (D,Q) = \frac{\sum_{k=1}^{t} w_{qk}\ .\ w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2 .\ \sum_{k=1}^{t} (w_{dk})^2}}$$

Où :

$\sum_{k=1}^{t} w_{qk}\ .\ w_{dk}$: le produit vectoriel entre la requête et le document, noté par le produit : D.Q

$\sqrt{\sum_{k=1}^{t} (w_{qk})^2 .\ \sum_{k=1}^{t} (w_{dk})^2}$: la norme de vecteur document et requête est respectivement noté par : $\|D\|.\|Q\|$

L'équation de similarité devient :

$$similarite\ (D,Q) = \frac{\mathrm{D.Q}}{\|D\|.\|Q\|}$$

3.4 TF-IDF :

Les termes peuvent être pondérés en fonction de leur importance. La méthode de pondération TF-IDF (Term Frequency-Inverse Document Frequency) est une méthode fréquemment utilisée en recherche d'information. Elle permet de calculer un degré d'importance à chaque terme contenu dans un document, relativement à un corpus.[38]

3.4.1 Fréquence du terme :

La fréquence d'un terme (TF : Term Frequency) est le pourcentage d'apparition de ce terme dans un document. Soit D : un document et T : un terme, la fréquence du terme T dans le document D, et on la note par $tf_{T,D}$, est calculée par cette formule :

$$tf_{T,D} = \frac{n_{T,D}}{\sum_{terme} n_{terme,D}}$$

Où :

$n_{T,D}$: le nombre d'occurrence de terme T dans le document D.

$\sum_{terme} n_{terme,D}$: La somme des occurrences de tous les termes qui apparaissent dans le document D.[39]

Exemple :

Document 1
C'est une vaste péninsule couverte en partie de déserts et baignée par trois mers : la mer Rouge à l'occident, la mer d'Oman et le golfe Persique à l'orient, la mer des Indes au midi.

Tableau 3.2 : document tiré de livre « La civilisation des Arabes (1884) »

On peut calculer, par exemple, la fréquence du terme « mer » dans le document illustré dans le tableau précédent (la ponctuation et l'apostrophe sont ignorées). L'occurrence du terme « mer » est : $n_{\text{mer},Document} = 3$.

Vingt-six termes apparaissent une fois. Trois termes (et, à, l) apparaissent deux fois. Deux termes (la, mer) apparaissent trois fois. Donc, la valeur de dénominateur sera :

$$\sum_{terme} n_{terme,D} = 26 + 3*2+2*3 = 38$$

Par substitution dans la formule, on obtient : $tf_{\text{mer},Document} = \frac{3}{38} = 0{,}0789$

3.4.2 Fréquence inverse de document :

La fréquence du terme, présentée ci-dessus souffre d'un problème critique : tous les termes sont considérés de la même importance lors de l'évaluation de la pertinence d'une requête.[38]

La fréquence inverse de document IDF (inverse document frequency) sert à évaluer l'importance d'un terme dans une collection de documents. Elle vise à attribuer un poids essentiel aux termes les moins fréquents observés comme plus éliminateurs. Elle consiste à calculer le logarithme de l'inverse de la proportion de leur collection qui contiennent le terme. [36, 37]

$$idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

$|D|$: nombre total de documents dans la collection de documents.

$|\{d_j : t_i \in d_j\}|$: nombre de documents ou le terme t_i apparait.

Enfin, le poids tf-idf s'obtient en multipliant les deux mesures : fréquence du terme (TF) et fréquence inverse de documents (IDF)

$$tfidf_{i,j} = tf_{i,j} * idf_j$$

Exemple :

Document 1	**Document 2**	**Document 3**
Les maisons des populations **Arabes** des classes moyennes et inférieures sont parfois d'une simplicité extrême et diffèrent beaucoup, à ce point	Leur disposition générale est la même dans tout l'Orient ; mais dans les pays où se fait sentir l'influence européenne, elles perdent beaucoup de leur	C'est ainsi que toutes les maisons des **Arabes** des bords du Nil sont construites uniquement avec le

de vue, des maisons luxueuses des **Arabes** aisés que nous décrirons dans notre prochain chapitre.	style primitif. Palmiers, ont un aspect oriental si caractéristique.	limon de ce fleuve. Elles sont édifiées simplement avec des

Tableau 3.3 : document tiré de livre « La civilisation des Arabes (1884) »

A tire d'exemple, on va choisir le document 1, soit d_1, et le terme analyse est « Arabes », soit t_1 = « Arabes ». La ponctuation et l'apostrophe sont ignorées.

Donc la formule devient :

$$idf_1 = log \frac{|D|}{|\{d_1 : t_1 \in d_1\}|}$$

Où $|D|$: est le nombre total de documents dans la collections de document, dans notre exemple on a trois documents $|D|$=3.

Le terme « Arabes » apparait dans le premier document et le troisième document. Ainsi :

$$idf_1 = log \frac{|D|}{|\{d_1 : t_1 \in d_1\}|} = log \frac{3}{2}$$

Application :

Tout au long de cette application nous allons montrer, à travers un exemple, l'utilité du modèle vectoriel. Considérant une collection, C, des documents tirés du livre « La civilisation des Arabes (1884) » illustré dans le tableau ci-dessous et la requête Q = 'Orient Occident civilisation '. Quelle sera la liste ordonnée de ces documents qui répondra le mieux à la requête Q ?.

Document 1	Document 2	Document 3
La civilisation des Arabes fut créée par un peuple à demi barbare.	Le contraste entre l'Orient et l'Occident est aujourd'hui trop grand,	C'est de l'Orient que l'Occident est né, et c'est encore à l'Orient qu'il faut aller demander

Tableau 3.4 : documents tirés de livre « La civilisation des Arabes (1884) ».

1.Les valeurs idf :

Le nombre total de documents est $|D|$=3. Certains termes apparaissent dans deux documents, d'autres dans un seul. Par exemple, le terme 'La' apparait une seule fois dans le document 1, dans ce cas, la valeur idf sera : $idf = log_2 \frac{3}{1}$ = 1,58. Ainsi, la liste suivante illustre les valeurs idf pour chaque terme.

La : log_2 3/1 =1,58	**Le** : log_2 3/1 =1,58
civilisation : $log_2$3/1 =1,58	**contraste** : log_2 3/1 =1,58
des : log_2 3/1 = 1,58	**l** : log_2 3/2 =0,58
Arabes : log_2 3/1 =1,58	**orient** : log_2 3/2 =0,58
Fut : log_2 3/1 =1,58	**et** : log_2 3/1 =1,58
créée : log_2 3/1 =1,58	**occident** : log_2 3/2 =0,58
par : log_2 3/1 =1,58	**est** : log_2 3/2 =0,58
un : log_2 3/1 =1,58	**aujourd** : log_2 3/1 =1,58
peuple : log_2 3/1 =1,58	**hui** : log_2 3/1 =1,58
à : log_2 3/2 =0,58	**trop** : log_2 3/1 =1,58
grand : log_2 3/1 =1,58	**aller** : log_2 3/1 =1,58
c : log_2 3/1 =1,58	**demander** : log_2 3/1 =1,58
de : log_2 3/1 =1,58	**entre** : log_2 3/1 =1,58

que : log_2 3/1 =1,58	**demi** : log_2 3/1 =1,58
né : log_2 3/1 =1,58	**barbre** : log_2 3/1 =1,58
encore : log_2 3/1 =1,58	
qu : log_2 3/1 =1,58	
Il : log_2 3/1 =1,58	
faut : log_2 3/1 =1,58	

Tableau 3.5 : Les valeurs idf de chaque terme

2**. Les valeurs tf :**

Ensuite, pour tous documents on calculera les fréquences pour tous les termes de la collection C. Ainsi, les trois listes suivantes illustrent les valeurs de fréquence des termes dans chaque document.

La : 1	**Le** : 0
civilisation : 1	**contraste** : 0
des : 1	**l** : 0
Arabes : 1	**orient** : 0
Fut : 1	**et** : 0
créée : 1	**occident** : 0
par : 1	**est** : 0
un : 1	**aujourd** : 0
peuple : 0	**hui** : 0
à : 1	**trop** :0

grand : 0	**aller** : 0
c : 0	**demander** : 0
de : 0	**entre** : 0
que : 0	**demi** : 1
né : 0	**barbre** 1
encore : 0	

qu : 0	
Il : 0	
faut : 0	

Tableau 3.6 : Les fréquences de tous les termes dans le document 1

La : 0	**Le** : 1
civilisation : 0	**contraste** : 1
des : 0	**l** : 1
Arabes 0	**orient** : 1
Fut : 0	**et** : 1
créée : 0	**occident** : 1
par : 0	**est** : 1
un : 0	**aujourd** : 1
peuple : 0	**hui** : 1
à : 0	**trop** :1
grand : 1	**aller** : 0
c : 0	**demander** : 0
de : 0	**entre** : 0
que : 0	**demi** : 0
né : 0	**barbre** : 0
encore : 0	
qu : 0	
Il : 0	
faut : 0	

Tableau 3.7 : Les fréquences de tous les termes dans le document 2

La : 0	**Le** : 0
civilisation : 0	**contraste** : 0
des : 0	**l** : 2
Arabes : 0	**orient** : 2
Fut : 0	**et** : 1
créée : 0	**occident** : 1
par : 0	**est** : 1
un : 0	**aujourd** : 0
peuple 0	**hui** : 0
à : 1	**trop** :0
grand : 0	**aller** : 1
c : 1	**demander** : 1
de : 1	**entre** : 0
que : 1	**demi** : 0
né : 1	**barbre** : 0
encore : 1	
qu : 1	
Il : 1	
faut : 1	

Tableau 3.8 : Les fréquences de tous les termes dans le document 3

3.Les valeurs idf x tf :

Après avoir calculé les valeurs de fréquences (tf) ainsi que les fréquences inverses de chaque terme, nous allons multiplier les valeurs tf par les valeurs idf de chacun. Selon notre exemple, le terme 'La' apparait une seule fois dans le document 1, ainsi sa fréquence égale à tf =1 et sa fréquence inverse égale à idf= 1,58.Donc, par

multiplication de ces deux dernières valeurs on obtient idf x tf = 1 x 1,58=1,58**.** Voici un tableau de documents-en-termes illustrant tous les résultats obtenus.

Document 1	**La** : 1,58 **civilisation** : 1,58 **des** : 1,58 **Arabes** : 1,58 **Fut :** 1,58 **créée** : 1,58 **par** : 1,58 **un :** 1,58 **peuple :** 0	**à** : 0,58 **grand** : 0 **c** : 0 **de** : 0 **que** : 0 **né** : 0 **encore** : 0 **qu** : 0 **Il** : 0	**faut** :0 **Le** : 0 **contraste** : 0 **l** : 0 **orient** : 0 **et** : 0 **occident** : 0 **est** : 0	**aujourd** : 0 **hui** : 0 **trop** :0 **aller** : 0 **demander** : 0 **entre** : 0 **demi** : 1,58 **barbre** : 1,58
Document 2	**La** : 0 **civilisation** : 0 **des** : 0 **Arabes** : 0 **Fut :** 0 **créée** : 0 **par** : 0 **un :** 0 **peuple :** 0	**à** : 0 **grand** : 1,58 **c** : 0 **de** : 0 **que** : 0 **né** : 0 **encore** : 0 **qu** : 0 **Il** : 0	**faut** : 0 **Le** : 1,58 **contraste** : 1,58 **l** : 1,58 **orient** : 0,58 **et** : 1,58 **occident** : 0,58 **est** : 0,58	**aujourd** : 0 **hui** : 0 **trop** :0 **aller** : 1,58 **demander** :1,58 **entre** : 0 **demi** : 0 **barbre** : 0

Document 3	**La** : 0	**à** : 0,58	**faut** : 1,58	**aujourd** : 0
	civilisation : 0	**grand** : 0	**Le** : 0	**hui** : 0
	des : 0	**c** : 1,58	**contraste** : 0	**trop** :0
	Arabes : 0	**de** : 1,58	**l** : 1,16	**aller** : 1,58
	Fut : 0	**que** : 1,58	**orient** : 1,16	**demander** :1,58
	créée : 0	**né** : 1,58	**et** : 1,58	**entre** : 0
	par : 0	**encore** : 1,58	**occident** : 1,58	**demi** : 0
	un : 0	**qu** : 1,58	**est** : 0,58	**barbre** : 0
	peuple : 0	**Il** : 1,58		

Tableau 3.9 : Valeurs idf x tf pour chaque document

4. Le vecteur tf x idf pour la requête :

Maintenant nous allons calculer le vecteur tf x idf da la requête Q = 'Orient Occident civilisation '. Nous multiplions la fréquence de chaque terme de Q (par exemple la fréquence du terme Orient dans Q égale 1) par sa valeur idf correspondante (voir Tableau 3.5). Par exemple, la valeur tf x idf du terme 'civilisation' égale à : $tf\ x\ idf = 1\ x\ 1{,}58\ =\ 1{,}58$. Ci-après, le vecteur tf x idf pour la requête Q = 'Orient Occident civilisation '.

Requête Q	**La** : 0	**à** : 0	**faut** :0	**aujourd** : 0
	civilisation : 1,58	**grand** : 0	**Le** : 0	**hui** : 0
	des : 0	**c** : 0	**contraste** : 0	**trop** :0
	Arabes : 0	**de** : 0	**l** : 0	**aller** : 0
	Fut : 0	**que** : 0	**orient** : 0,58	**demander** : 0
	créée : 0	**né** : 0	**et** : 0	**entre** : 0
	par : 0	**encore** : 0	**occident** : 0,58	**demi** : 0
	un : 0	**qu** : 0	**est** : 0	**barbre** : 0
	peuple : 0	**Il** : 0		

Tableau 3.10 : Valeurs idf x tf pour la requête

5. Le calcul de similarité :

A la fin, nous calculons la similarité entre chaque document et la requête. On sait que la formule de similarité est donnée par : $similarite\ (D,Q) = \frac{\text{D.Q}}{\|D\|.\|Q\|}$ dont :

D.Q : le produit vectoriel entre la requête et le document.

$\|D\|.\|Q\|$: le produit de la norme de vecteur requête et vecteur document

Ainsi, les valeurs des normes de chaque vecteur sont :

$\|Document\ 1\|$

$= \sqrt{(1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (0{,}58)^2 + (1{,}58)^2 + (1{,}58)^2}$

$= 5{,}029$

$\|Document\ 2\|$

$= \sqrt{(1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (0{,}58)^2 + (1{,}58)^2 + (0{,}58)^2 + (0{,}58)^2 + (1{,}58)^2 + (1{,}58)^2}$

$= 4{,}26$

$\|Document\ 3\|$

$$= \sqrt{\begin{matrix}(0{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}58)^2 + (1{,}16)^2 + \\ (1{,}16)^2 + (1{,}58)^2 + (1{,}58)^2 + (0{,}58)^2 + (1{,}58)^2 + (1{,}58)^2\end{matrix}}$$

$= 5{,}772$

$\|Q\| = \sqrt{(1{,}58)^2 + (0{,}58)^2 + (0{,}58)^2} = 1{,}78$

Ensuite, les valeurs de similarité sont :

$similarite\ (Document1, Q)$

$$= \frac{\begin{matrix}1{,}58 * 0 + 1{,}58 * 1{,}58 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 0 * 0 + 0.58 * 0 + \\ 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0{,}58 + 0 * 0 \\ +0 * 0{,}58 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1{,}58 * 0 + 1{,}58 * 0\end{matrix}}{5{,}029 * 1{,}78}$$

$= 0{,}883$

$similarite\ (Document2, Q)$

$$= \frac{\begin{matrix}0 * 0 + 0 * 1{,}58 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + \\ 1{,}58 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + \\ 0{,}58 * 0{,}58 + 1{,}58 * 0 + 0{,}58 * 0{,}58 + 0{,}58 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 0 * 0 + 0 * 0 + \\ 0 * 0\end{matrix}}{4{,}26 * 1{,}78}$$

$= 0{,}281$

$similarite\ (Document3, Q)$

$$= \frac{\begin{matrix}0 * 0 + 0 * 1{,}58 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0.58 * 0 + \\ 0 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + 1{,}58 * 0 + \\ 0 * 0 + 0 * 0 + 1{,}16 * 0 + 1{,}16 * 1{,}58 + 1{,}58 * 0 + 1{,}58 * 0{,}58 + 0{,}58 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + \\ 1{,}58 * 0 + 1{,}58 * 0 + 0 * 0 + 0 * 0 + 0 * 0\end{matrix}}{5{,}029 * 1{,}78}$$

$= 1{,}413$

Selon les valeurs de similarité, les documents sont présentés comme résultat à la requête dans cet ordre : document 3, document 1, document 2

3.5 Avantages :

Par rapport au modèle standard booléen, le modèle vectoriel présente les avantages suivants : [37, 40] :

1. Les poids des termes ne sont pas binaires.

2. C'est un modèle simple qui se base sur l'algèbre linéaire.

3. Le modèle vectoriel permet de calculer un degré continu de similarité entre les requêtes et les documents.

4. Il classe les documents en fonction de leur pertinence.

3.6 Limitation :

Le modèle vectoriel présente les limitations suivantes [37, 40] :

1.Les valeurs de similarité des documents longs sont pauvres, par conséquent les documents sont mal représentés.

2.La recherche des mots clés doit correspondre précisément au terme du document ; une sous chaine de caractères pourrait se traduire par une « fausse correspondance positif"

3. La sensibilité sémantique ; documents avec un contexte similaire, mais avec un vocabulaire différent ne seront pas associés, ce qui entraîne une "fausse correspondance négative".

4. L'ordre dans lequel les termes apparaissent dans le document est perdu dans la représentation de l'espace vectoriel.

5.Theoriquement les termes considérés sont statistiquement indépendants.

6.La pondération est intuitive mais pas très formelle.

3.7 Conclusion :

Dans ce chapitre, nous avons présenté les concepts mathématiques du modèle vectoriel, le schéma de pondération TF-IDF, ainsi que ses avantages et ses inconvénients.

Le chapitre suivant mettra en lumière la définition des règles d'association leurs concepts de base ainsi que la présentation de l'algorithme Apriori, qui permet l'extraction de ce type de règle, et son déroulement sur un exemple concret. De plus, nous présentons les inconvénients et les avantages de ces règles.

Chapitre 4

Les règles d'association

4.1 Introduction :

Une quantité importante de données s'accumulent lors des opérations quotidiennes dans des entreprises commerciales. A tire d'exemple, le tableau 4.1 illustre la nature de données collectées par les achats des consommateurs dans les grands magasins. Chaque ligne de la table correspond à une transaction identifiée par un numéro (TID) ainsi que les produits achetés. Pour mieux comprendre le comportement des clients, les commerçants s'intéressent à l'analyse de données de chacun.[41]

TID	Items
1	Pain, Lait
2	Pain, Couches, Bière, Œufs
3	Lait, Couches, Bière, Coca
4	Pain, Lait, Couches, Bière
5	Pain, Lait, Couches, Coca

Tableau 4.1 : achats des consommateurs

Le tableau 4.1, montre qu'il existe une relation forte entre la vente des couches et la vente de bière. Par conséquent, les clients qui achètent les couches achètent aussi de la bière.

On représente cette relation par une règle d'association :

Couches → Bière.

D'une manière générale on représente une règle d'association par :

Antécédent⟶ Conséquent

Cette règle est lue comme suit : si une condition existe, alors forcément, un résultat issu de celle-ci existe aussi.[4, 41]

Introduites par Hajek dans les années 60, les règles d'associations représentent un domaine relativement récent. Dans ce chapitre, nous allons expliquer les concepts de base de l'analyse d'associations et l'algorithme associé.

4.2. Définitions :

Dans la section suivante nous allons détailler plusieurs concepts impliqués dans la recherche et l'extraction des règles d'association à savoir : transaction, item, support, confiance, règle d'association.[3, 41-43]

4.2.1. Transaction et ensemble d'items :

Soient $\Gamma = \{I_1, I_2, I_3, \dots, I_m\}$ l'ensemble d'attributs binaires appelés items. Soit T une base de données des transactions. Chaque transaction t est représentée comme un vecteur binaire, avec $t[k] = 1$ si la transaction t achète l'item I_k, sinon $t[k] = 0$ .Chaque transaction est définie par un seul numéro dans la base des transactions.

Exemple :

Soient $\Gamma = \{\text{Pain}, \text{Lait}, \text{Couches}, \text{Bière}, \text{Œufs}, \text{Coca}\}$, l'ensemble de tous les items des paniers et $D = \{1,2,3,4,5\}$ l'ensemble de toutes les transactions. Le tableau 4.2 pourrait se mettre sous forme binaire comme suit :

TID	Pain	Lait	Couches	Bière	Œufs	Coca
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1

4	1	1	1	1	0	0
5	1	1	1	0	0	1

Tableau 4.2 : tableau des transactions présentés en binaires

4.2.2. Itemset, Itemset fréquent et support :

Soient $\Gamma = \{I_1, I_2, I_3, \dots, I_m\}$ l'ensemble de tous les items. L'ensemble I est un Itemset si et seulement si $I \subseteq \Gamma$. Un K-Itemset est un Itemset de k Items. Un Itemset est fréquent si et seulement si son support est supérieur à un support minimum.

Un concept fondamental pour un itemset est son support qu'on définit par le nombre de transactions qui le contient, divisé par le nombre total des transactions de D. Mathématiquement, le support $\sigma(A)$ d'un itemset est défini par :

$$\sigma(A) = \frac{Card\ (A)}{Card(D)}$$

Exemple :

D'où, dans le tableau 4.1 le support {Couches, Bière} est égal à 3/5.

4.2.3 Règle d'association, support et confiance :

Soient A et B deux sous-ensembles d'items disjoints. Une règle d'association est de la forme $A \rightarrow B$ possède deux métriques importantes pour mesurer sa force à savoir : le support et la confiance.

Le support d'une règle d'association $A \rightarrow B$ est le support $\sigma(A \cup B)$, divisé par le nombre total des transactions de D.

$$support\ , s(A \rightarrow B) = \frac{Card(A \cup B)}{Card(D)}$$

La confiance d'une règle d'association $A \rightarrow B$ est le support $\sigma(A \cup B)$, divisé par le support $\sigma(A)$.

$$confiance, c(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(A)}$$

Exemple :

Considérons la règle {Pain, Lait} → {Couches}. La $Card$ ($\{Pain,$ lait, Couches$\}$) étant égal à 2 et le nombre de transactions étant égal à 5, le support de la règle est 2/5=0,4. La confiance de la règle est la division entre le support $support$ ($\{Pain,$ lait, Couches$\}$) et le $support(\{Pain,$ lait$\})$ qui est égale à 3/5=0,6. Ainsi, la confiance de cette règle est 0,4/0,6=0,66.

4.3. La recherche des règles d'association :

On peut classer la recherche des règles d'association par les trois étapes suivantes : [6, 10]

1. Préparation des données :

La préparation des données est une étape primordiale. Elle vise à réduire la quantité des données tout en gardant seulement les plus pertinentes. Ensuite, ces dernières seront transformées en un contexte d'extraction, autrement dit elles formeront des triplets constitués :

- d'un ensemble fini d'items.

- d'un ensemble fini d'objets.

- d'une relation binaire entre ces deux ensembles.

Exemple :

Soient $I = \{a, b, c, d, e, f\}$ un ensemble d'items et $O = \{1,2,3,4,5\}$ un ensemble d'objets tels que présentés à la figure 4.1. Soit R une relation binaire entre les items de l'ensemble I et les objets de l'ensemble O. Chaque couple de la relation binaire , associant un item $i \in I$ et un objet $o \in O$ est représenté par : oRi.

Items / Objets	a	b	c	d	e	f
1	x	x				
2			x	x	x	
3		x	x	x		x
4	x	x	x	x		
5	x	x	x			x

Figure 4.1 : l'ensemble des items et des objets.

2. Recherche des Itemsets fréquents :

Cette étape est très coûteuse en temps d'exécution. Pour un ensemble d'items, le nombre d'Itemsets fréquents qui peut être générés est de 2^n .

3. Production des règles d'association :

La génération des règles d'association consiste à trouver toutes les règles ayant un support $\geqslant minsup$ et une confiance $\geqslant minconf$ où $minsup, minconf$ sont des seuils, définis pas l'utilisateur, pour le support et la confiance respectivement. Cette étape est aussi, coûteuse en temps d'exécution. Par exemple pour un ensemble de d items le nombre total de règles possibles est : $R = 3^d - 2^{d+1} + 1$.

4.4. L'algorithme Apriori :

Proposé par Agrawal et Srikant en1994, l'algorithme Apriori représente la base de tous les algorithmes de recherche des règles d'association. Il extrait les Itemsets fréquents pour les règles d'association.[3, 44]

4.4.1 Le principe de l'algorithme Apriori :

L'algorithme Apriori utilise une approche itérative, où $k-Itemsets$ sont employés pour explorer les $(k+1)-Itemsets$. D'abord, les $1-Itemsets$ sont trouvés par balayage de la base de données pour calculer le support de chaque item, et la collecte de ces Itemsets qui ont un support $\geqslant minsup$. L'ensemble résultant est noté L_1, puis utilisé pour trouver L_2 , les 2-itemsets, qui est utilisé pour trouver L_3 , et ainsi de suite jusqu'à ce qu'aucun k-Itemsets puisse être trouvé. L'obtention de chaque L_k nécessite une analyse complète de la base de données.[44]

4.4.2 L'algorithme Apriori :

La description complète de l'algorithme Apriori se résume dans les étapes suivantes [3]:

Input : un support minimum et une base de données de transactions.

Output : génération des Itemsets fréquents

1. $M_i = \phi, i = 0$
2. C_1= tous les 1-Itemsets dans la base de données
3. L_1 = tous les Itemsets fréquents de C_1

Tant que (M_i est non vide) **Faire**

1. C_{i+1} = Candidate-gen (L_i)
2. L_{i+1}= tous les Itemsets fréquents de C_{i+1}
3. $i++$
4. Retourner l'union des M_i

Fin tant que

Algorithme 4.1 : l'algorithme Apriori

Exemple :

Considérant les données de tableau 4.2 il y a neuf transactions dans cette base de données, soit, $|D| = 5$. Nous utilisons la figure 4.1 pour illustrer l'algorithme Apriori pour trouver Itemsets fréquents dans D. Supposons que le support minimum requis est égal à 2, soit, $minsup = 2$.

1. Dans la première itération de l'algorithme (figure 4.1), chaque item appartient à l'ensemble 1-itemsets, C_1. Ensuite, l'algorithme calcule l'occurrence de chaque élément de C_1. Après avoir calculer les occurrences, il compare chaque valeur avec la valeur $minsup$ afin d'éliminer les items qui ne satisfont pas le $insup$. Le résultat obtenu détermine l'ensemble fréquents, L_1 .

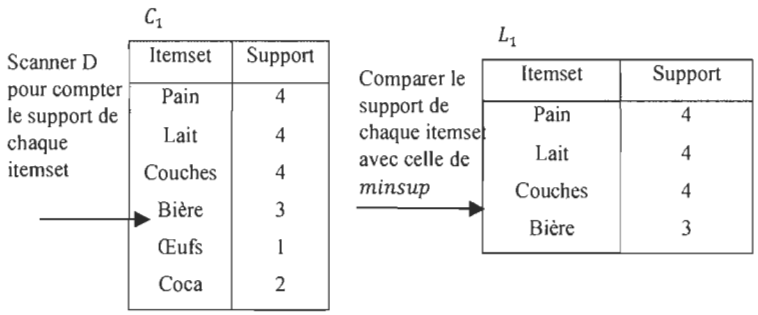


Figure 4.2 : la première itération de l'algorithme Apriori

2. Dans la deuxième itération (figure 4.2), l'algorithme utilise la jointure $L_1 \bowtie L_1$ pour générer l'ensemble des candidats 2-itemsets, C_2.Ensuite, il analyse les transactions dans le but de calculer le support de chaque candidat en C_2 . Enfin, l'algorithme supprime les candidats avec lequel leur support est inferieur a $minsup$ pour obtenir l'ensemble fréquents, L_2.

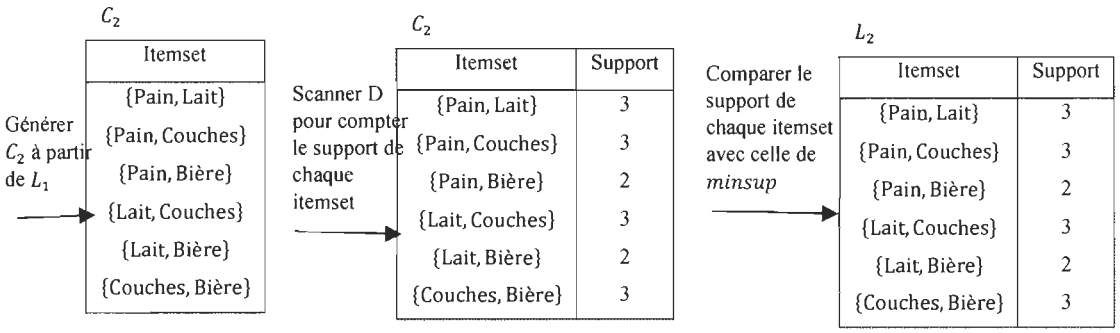


Figure 4.3 : la deuxième itération de l'algorithme Apriori

3. Dans la troisième itération (figure 4.3), la génération de C_3 est faite par la jointure $L_1 \bowtie L_2$. L'algorithme scanne les transactions pour calculer l'occurrence de chaque item. L'ensemble fréquents, L_3, est déterminé par l'élimination des candidats qui ne satisfont pas le *minsup*.

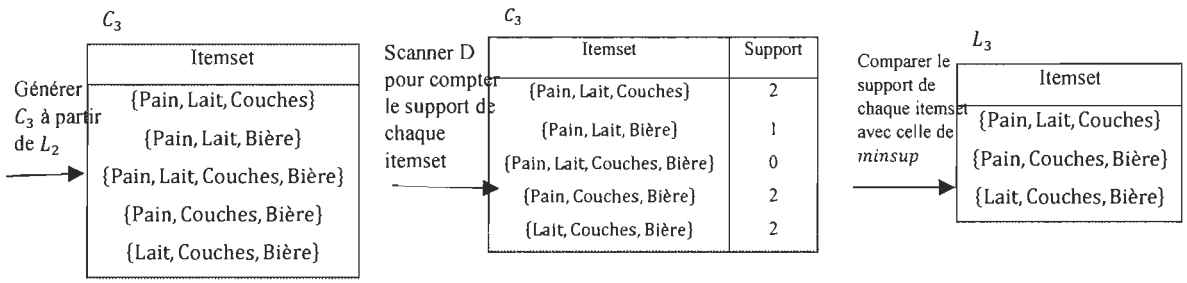


Figure 4.4 : la troisième itération de l'algorithme Apriori

4. Enfin, L'algorithme calcule l'ensemble C_4 par l'utilisation de jointure $L_3 \bowtie L_3$. Le résultat de C_4 est constitué d'un seul candidat, à savoir {Pain, Lait, Couches, Bière}, qu'il ne satisfait pas le *minsup* Ainsi l'algorithme se termine, à ce stade-là, en affichant tous les Itemsets fréquents trouvés.

4.4.3 Générer les règles d'association à partir d'Itemsets fréquents :

Une règle d'association forte satisfait à la fois le *minsup* et le *minconf*. Les règles d'association peuvent être générées comme suit [44] :

1. Pour chaque itemset fréquent , générer tous les sous-ensembles non vides de l.

2. Pour tout sous-ensemble non vide s de l, la règle "$s \Rightarrow (l - s)$" est générée si le support de $(l - s)$ divisé par le support de s est supérieur ou égale à $minsup$.Où, $(l - s)$ est l'ensemble des éléments qui appartiennent à l mais pas a s.

Exemple :

Considérons les données de tableau 4.2 Supposons que les données contiennent l'itemset fréquent l = {Pain, Couches, Bière}. Les sous-ensembles non vides de l : {Pain, Couches}, {Pain, Bière}, {Couches, Bière}, {Pain}, {Couches}, et {Bière}. Les règles d'association résultants sont indiquées ci-dessous :

1. {Pain, Couches} → {Bière}, confiance = 66 %
2. {Pain, Bière} → {Couches}, confiance = 100 %
3. {Couches, Bière} → {Pain}, confiance = 66 %
4. {Pain} → {Couches, Bière}, confiance = 50 %
5. {Couches} → {Pain, Bière}, confiance = 50 %
6. {Bière} → {Pain, Couches}, confiance = 66 %

Si le $minsup$ par exemple est de 60%, alors la première, la deuxième, la troisième, et la dernière règle seront affichées en sortie.

4.5 Avantages :

On peut résumer les avantages des règles d'association dans :

1. La possibilité de découverte des connaissances utiles, cachées dans les bases de données [3]

2. Leurs facilités de compréhension, efficacité et simplicité. [3]

3. Leur formalisme non supervisé et général. [3]

4. Le forage des règles d'association est un grand succès dans divers domaines que ce soit dans des activités commerciales, sociales ou humaines. [45]

4.6 Inconvénients :

Quelques inconvénients des règles d'association :

1. La découverte d'un nombre important de règles d'association dont la plupart ne sont pas intéressantes. [46, 47]

2. Le temps de recherche des Itemsets fréquents est énorme. [3]

3. Les algorithmes utilisés ont trop de paramètres, par conséquent l'extraction de données, pour les non experts, devient compliquée. [47]

4. Un problème de sécurité pourrait être posé : des renseignements confidentiels peuvent être facilement divulgués, en utilisant cette technique. [45]

5. L'utilisation d'un seul *minsup* pourrait engendrer un dilemme d'item rare ; celui-ci signifie que tous les éléments de la base de données sont de même nature. Ce qui n'est pas toujours vrai. [12]

4.7 Conclusion :

Dans ce chapitre, nous avons présenté les règles d'association, leurs concepts de base, l'algorithme Apriori et son déroulement sur un exemple concret. De plus, nous avons motionné les inconvénients et les avantages de cette approche.

Dans le chapitre suivant nous allons détailler l'architecture du système développé qui consiste en trois étapes : la création d'index inversé, la classification et l'extraction des règles d'association. Pour chaque étape, nous présenterons le processus de conception et les algorithmes associés.

Chapitre 5

Méthodologie

5.1 Introduction :

Ce mémoire traite la problématique de sélection de mots clés (key-termes), par l'emploi des méthodes de classifications et les règles d'association.

Le processus de classification appliqué ici est différent de ceux présentés dans les autres travaux [3] [4] qui font une classification sur les documents. Dans notre méthode, nous classifions les termes au lieu des documents, vu que la taille du vocabulaire est assez grande par rapport au nombre du document[48]. De plus, le classifieur utilise les vecteurs stockés dans une matrice préalablement bâtie, et basée sur le modèle TF-IDF. L'extraction des règles d'association à partir de classes résultantes s'avère efficace dans la sélection des mots clés d'un corpus.

Dans ce chapitre, nous verrons tout d'abord l'architecture globale de notre projet dans laquelle nous donnerons les différentes étapes. Nous détaillerons chaque étape tout en montrant l'objectif de celle-ci ainsi que ses algorithmes avec l'ensemble des justifications.

5.2 Architecture de notre système :

La figure 5.1 résume l'architecture générale de notre projet. Dans un premier temps, nous enregistrons les textes sous forme d'index inversé. Ce dernier est une structure d'index de données stockant un lien à partir des mots vers la liste de segments (paragraphes, phrases ou mots) où ils se trouvent (voir section 5.3.5). Le but d'un index inversé est de faciliter la recherche des mots, la création de la matrice TF-IDF ainsi que les opérations sur les règles d'association : union, intersection. Par la suite, nous utilisons la matrice TF-IDF générée par la première étape afin de produire des classes de

termes ; l'utilisateur choisit un classifieur parmi ceux qui sont déjà affichés dans la plateforme. Enfin, la troisième étape, quant à elle, utilise les classes résultantes de l'étape précédente pour extraire les règles d'associations.

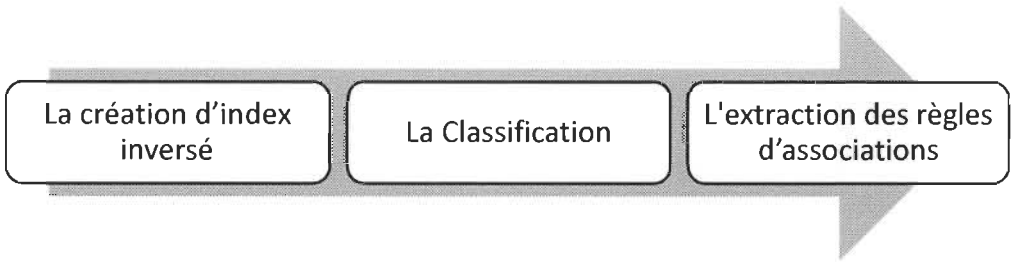


Figure 5.1 : architecture générale du projet

5.3 La création d'index inversé :

D'abord notre système reçoit, en entrée, différents formats de fichier : format HTML, format PDF et format Doc…etc. Ensuite, on extrait le texte brut de l'ensemble de ceux-ci, avec lequel on crée un index inversé. Nous pouvons diviser la phase de la création de l'index inversé en cinq étapes (figure 5.2) :

1. L'extraction de texte brut :

L'extraction de texte brut permet de récupérer de l'information pertinente depuis des fichiers en format particulière (PDF, Word, HTML, etc.). Pour parvenir à ces fins, nous devons récupérer que le contenu sans mise en forme et tous les autres éléments (figures, tableaux, organigrammes, etc.).

2. La segmentation :

La segmentation du corpus est une phase essentielle dans l'analyse du texte. Elle permet de découper le corpus en segments disjoints (paragraphes, phrases ou mots) dans le but de pouvoir les indexer par la suite.

3. L'extraction du vocabulaire :

L'extraction du vocabulaire a pour effet de créer des listes de mots à partir de tous les mots de segments. Pour arriver à ces fins, nous considérons les espaces, les apostrophes et les signes de ponctuations comme des délimiteurs des mots.

4. Le nettoyage du vocabulaire :

Le nettoyage du vocabulaire permet de supprimer certains mots non pertinents dans le but de réduire l'espace de stockage de l'index inversé et d'augmenter la performance de la classification en termes de temps d'exécution.

5. La construction de l'index inversé :

À partir de l'ensemble de mots nettoyés et les segments obtenus nous construisons l'index inversé qui sert à lier ces mots aux segments où ils se trouvent.

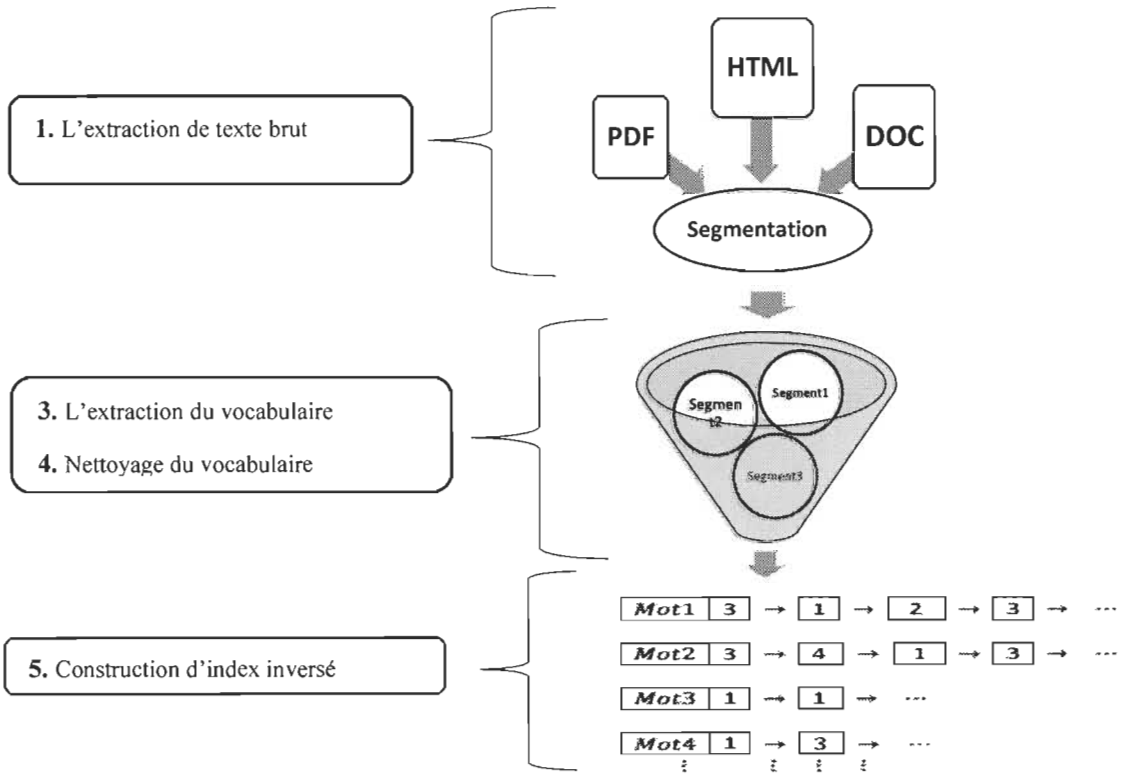


Figure 5.2 : processus de création d'index inversé

5.3.1 L'extraction de texte brut :

Nous récupérons le texte brut à partir de différents formats de fichiers grâce à la conversion de type. Lors de processus de récupération, nous devons éviter les ambigüités par exemple avec des balises HTML qui a serrent à décrire la structure du document à titre d'exemple la balise <h1> définit un grand titre.

5.3.2 Segmentation :

La segmentation sera appliquée sur le texte brut extrait dans le but d'être en mesure d'indexer les segments résultants et servir la création d'une matrice, dans laquelle ces segments représentent ses lignes, à des fins de classification. Le choix du type de la segmentation ainsi que le nombre d'éléments contenu dans chacun des segments relevés par l'utilisateur. Ce dernier devrait choisir l'une des façons suivantes :

- La segmentation en paragraphe(s).
- La segmentation en phrase(s).
- La segmentation en mot(s).
- La segmentation basée sur un mot particulier.
- La segmentation basée sur un signe particulier.

Algorithme :

Voici l'algorithme permettant de segmenter le texte.

Input : texte
Output : liste des segments
Début
Si Segmentation en paragraphe **Alors**
    Séparateur ← Retour chariot
Fin si
Si Segmentation en mot **Alors**
    Séparateur ← Espace

```
Fin si
Si Segmentation basée sur un mot particulier Alors
        Séparateur ← mot
Si segmentation basée sur un signe particulier Alors
        Séparateur ← signe
Fin si
Tant que (Séparateur non trouve) && (Non fin de texte) faire
        Chaine ← lire (mot)
        Ajouter_liste_element (Liste de segments, mot)
Fin Tant que
Fin
```

Algorithme 5.1 : l'algorithme de segmentation

5.3.3 Extraction du vocabulaire :

Nous construisons les listes des mots à partir des segments obtenus lors du processus de segmentation. Dans cette phase, nous considérons les espaces, les apostrophes et les lignes de ponctuations en tant que des séparateurs de mots. Le vocabulaire extrait subira certains nettoyages.

Algorithme :

Voici l'algorithme permettant l'extraction du vocabulaire.

```
Input : segment
Output : liste du vocabulaire
Début
Tant que Non fin de segment Faire
        motLu ← Lire un mot
        Procédure_Nettoyer ( motLu )
```

```
        Si motLu n'est pas supprimé Alors
            Ajouter motLu a la liste du vocabulaire
        Fin si
Fin Tant que
Fin
```

Algorithme 5.2 : l'algorithme pour l'extraction du vocabulaire

5.3.4 Nettoyage du vocabulaire :

La taille du vocabulaire extrait pourrait être assez grande, vu le volume important des documents en traitement, par conséquent l'espace de stockage de l'index inversé et le temps d'exécution du processus de classification pourraient augmenter. Donc, nous devons réduire la taille du vocabulaire en effectuant un certain nombre d'opérations pour éliminer plusieurs mots non pertinents pour les besoins de la classification. Nous avons par exemple les opérations suivantes :

- Lemmatisation du texte : Elle remplace les mots fléchis par leur forme canonique. Par exemple, un verbe conjugué est remplacé par le verbe à l'infinitif, le féminin est remplacé par le masculin, etc.
- Élimination des mots fonctionnels : Ils pourraient être des articles (de, des, les, ...) des pronoms (ses, moi,) ou de certains verbes (sont, seront).
- Élimination des mots dont la taille est inférieure à un seuil donné.

Algorithme :

Voici l'algorithme permettant l'extraction du vocabulaire.

```
Input : liste du vocabulaire, liste des mots fonctionnels, seuil
Output : liste du vocabulaire nettoyé
Début
Tant que Non fin de la liste vocabulaire Faire
```

Terme ← Tète (liste du vocabulaire)
Si terme ∈ (liste des mots fonctionnels) **Alors**
    Supprimer le terme de la liste du vocabulaire
Fin si
Si taille (terme) < seuil **Alors**
    Supprimer le terme de la liste du vocabulaire
Fin Si
Si Existe (table de lemmatisation, terme) **Alors**
    Ajouter le mot
Fin Si
Fin tant que
Fin

Algorithme 5.3 : l'algorithme pour le nettoyage du vocabulaire

5.3.5 L'index inversé :

La figure 5.3 montre la structure de l'index inversé. Nous affectons à chaque mot une liste de segments (paragraphes, phrases ou mots) où il se trouve, par exemple la liste 1,2,3 est affectée au $Mot1$. L'ensemble des mots est le vocabulaire obtenu lors de l'étape : extraction et nettoyage du vocabulaire. Dans la structure d'index, les mots sont ordonnés. Quant aux segments, ils sont tous extraits au moment de la segmentation. Chaque segment est identifié par un numéro unique. Nous enregistrons, également, la taille de chaque liste de segments, par exemple la taille de la liste du $Mot1$ qui égale à 3 est enregistrée.

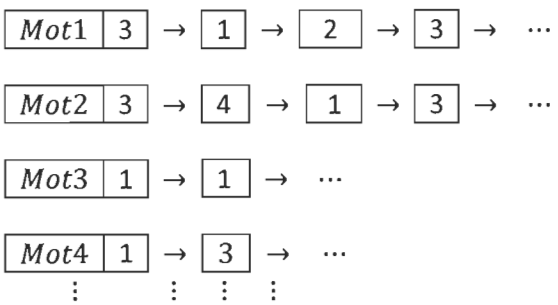
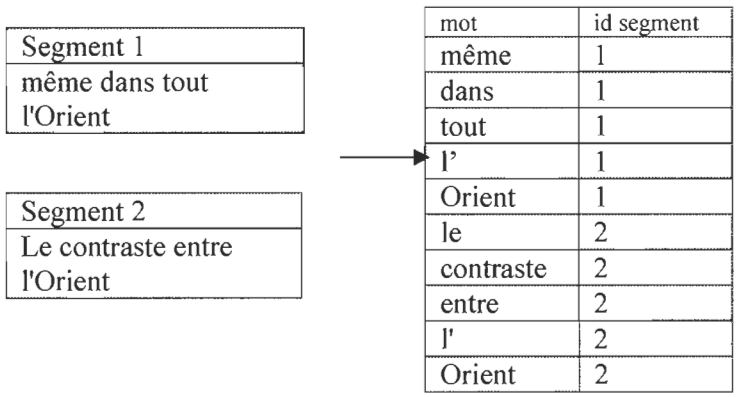


Figure 5.3 : structure d'index inversé

Durant la création de l'index inversé, nous construisons une liste de séquences de paires (mot, id segment) ; ainsi à chaque mot correspend son numéro de segments où il se trouve. Les mots dans la liste sont listés en respectant leur ordre d'apparition dans le segment.



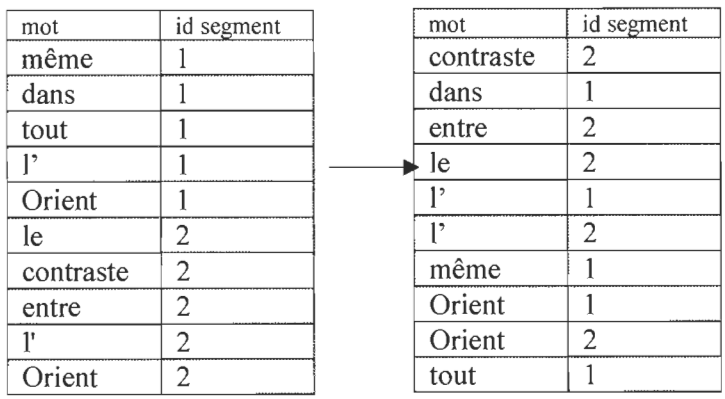
Segment 1
même dans tout l'Orient

Segment 2
Le contraste entre l'Orient

mot	id segment
même	1
dans	1
tout	1
l'	1
Orient	1
le	2
contraste	2
entre	2
l'	2
Orient	2

Figure 5.4 : liste de séquence de paires (mot, id segment)

La liste des paires (mot, id segment) sera ordonnée en fonction des mots. S'il existe une ressemblance entre au moins deux mots, nous les arrangeons en fonction de leur numéro de segments.



mot	id segment
même	1
dans	1
tout	1
l'	1
Orient	1
le	2
contraste	2
entre	2
l'	2
Orient	2

mot	id segment
contraste	2
dans	1
entre	2
le	2
l'	1
l'	2
même	1
Orient	1
Orient	2
tout	1

Figure 5.5 : liste des pairs (mot, id segment) ordonnée

Enfin, nous créons l'index inversé à partir des enregistrements résultants dans lequel chaque mot pointe vers la liste des segments. En outre, la taille de la liste sera ajoutée.

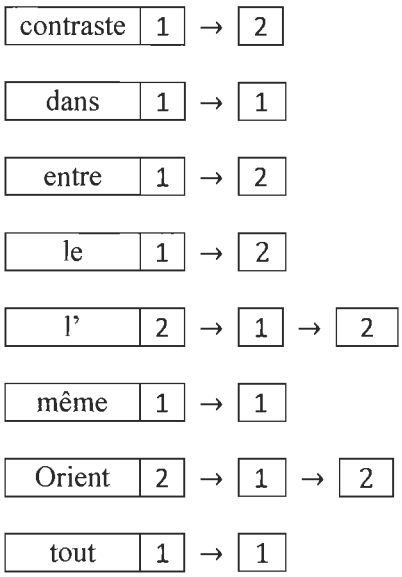


Figure 5.6 : création de l'index inversé

Algorithme :

Voici l'algorithme permettant la création d'index inversé.

Input : segments, table

Output : index inverse

Début

Liste_vocabulaire_nettoye ← **vide**

Pour Seg ∈ segments **Faire**

    Voc ← Extraire_Vocabulaire (Seg)

    Voc_nett ← Nettoyer_Vocabulaire (Voc)

    Ajouter_element (Liste_vocabulaire_nettoye , Voc_nett)

Fin pour

i ← 1

initialise_table (table , taille(Liste_vocabulaire_nettoye))

Pour ele ∈ Liste_vocabulaire_nettoye **Faire**

Table (i) ← (ele , numero_segment)

i ← i+1

Fin pour

Pour chaque ligne de la table **Faire**

1. Ajouter la liste des segments où se trouve le mot
2. Enregistrer la taille de chaque liste

Fin pour

Fin

Algorithme 5.4 : l'algorithme pour la création de l'index inversé

5.4. Classification :

La figure 5.7 montre le processus de classification. Afin de pouvoir appliquer un classifieur, nous devons utiliser la matrice TD-IDF comme entrée.

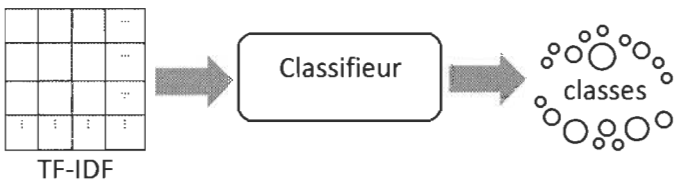


Figure 5.7 : processus de classification

5.4.1 La matrice TF-IDF :

Techniquement, on représente la matrice TF-IDF (tableau 5.1) sous forme d'un tableau de produits $TF * IDF$ à deux dimensions, où les lignes représentent les segments produits par l'étape de segmentation et les colonnes, ainsi que le vocabulaire nettoyé du texte à classifier.

	Terme1	Terme2	Terme3	Terme4	Terme5	...
Segment1	0,157	0,5	1,5	0,15	0,19	...
Segment2	1,758	0,999	0,145	1,33	0,16	...
Segment3	1,0	2,41	0	2,0	0,178	...
Segment4	1,3	0,5	0	0,20	1,0	...
...	...	...	...	...	...	...

Tableau 5.1 : tableau de produits $TF * IDF$

Pour construire la matrice TF-IDF, nous devons utiliser l'index inversé. Celui-ci, nous permet de calculer les fréquences des termes (TF) ainsi que la fréquence inversée des documents (IDF). Ainsi, Nous pouvons bâtir la matrice en question en deux étapes.

1. Calcul de fréquence des termes (TF) : Pour chaque terme de l'index on extrait la liste des segments correspondants. Ensuite, on calcule la fréquence du mot dans chaque élément de la liste en l'occurrence dans chaque segment.

terme	segment	fréquence du terme (TF)
contraste	2	0,4
...	...	...

Tableau 5.2 : fréquence des termes

2. Calcul de fréquence inverse de documents (IDF) : Afin de calculer la fréquence inversée du document, on doit utiliser la valeur enregistrée lors de création d'index qui correspond à la taille d'une liste. Celle-ci signifie le nombre de segments que contient le terme.

Exemple :

Considérons les segments de la figure 5.4 et la structure de l'index inversé correspondante telle que présentée à la figure 5.6. Nous allons calculer les

fréquences de chaque terme. La liste des segments du terme « contraste » est : 2. Le nombre d'occurrence du terme « contraste » dans le segment 2 égale 1. Ainsi, la fréquence du terme « contraste » égale $\frac{1}{5} = 0{,}2$. De la même manière nous calculons les fréquences des autres termes. Ainsi, le tableau suivant illustre les fréquences pour chaque terme.

	contraste	dans	entre	le	l'	même	orient	tout
Segment 1	0	0,2	0	0	0,2	0,2	0,2	0,2
Segment 2	0,2	0	0,2	0,2	0,2	0	0,2	0

Tableau 5.3 : les fréquences de tous les termes.

Le nombre total de segments est $|D| = 2$. Maintenant nous calculons les fréquences inverses de segments. Par exemple, le terme « contraste » apparait une seule fois dans le segment 1, dans ce cas, la valeur idf sera : $idf = log_2 \frac{2}{1} = 1$. Ainsi, le tableau suivant illustre les valeurs idf pour chaque terme.

	contraste	dans	entre	le	l'	même	orient	tout
idf	1	1	1	1	0	1	0	1

Tableau 5.4 : les valeurs idf de chaque terme

Le tableau suivant illustre les valeurs de $TF * IDF$ pour chaque segment.

	contraste	dans	entre	le	l'	même	orient	tout
Segment 1	0	0,2	0	0	0	0,2	0	0,2
Segment 2	0,2	0	0,2	0,2	0	0	0	0

Tableau 5.5 : les valeurs idf x tf pour chaque segment

Algorithme :

Voici l'algorithme permettant la création de la matrice TF-IDF.

```
Input : segment
Output : liste du vocabulaire
Début
Tant que Non fin de segment Faire
        motLu ← Lire un mot
        Procédure_Nettoyer ( motLu )
        Si motLu n'est pas supprimé Alors
            Ajouter motLu à la liste des mots sélectionnés
        Finsi
Fin Tant que
Fin
```

Algorithme 5.5 : création de la matrice TF-IDF

5.4.2 Choix du classifieur et le processus de classification :

Un classifieur regroupe des vecteurs ayant des similitudes dans des groupes homogènes. Le choix du classifieur revient à l'utilisateur qui en choisit un, parmi ceux qui sont offerts dans la plateforme à savoir : SOM, ART, KNN, K-mean, SVM. Nous avons déjà discuté de ces derniers dans le chapitre précédent.

Le classifieur choisi recevra, en entrée, la matrice TF-IDF, voir la section 5.4.1. Puis, à partir de celle-ci, il devra extraire les vecteurs pour les classifier. Pour cela, il considère que chaque colonne représente un vecteur. Autrement dit, il classifie les termes au lieu des segments. La dimension de chaque vecteur est déterminée par le nombre de segments du texte à analyser tandis que le nombre du vecteurs est en fonction de la taille du vocabulaire (figure 5.8).

	V1	V2	V3	...		
	Terme1	Terme2	Terme3	Terme4	Terme5	...
Segment1	0,157	0,5	1,5	0,15	0,19	...
Segment2	1,758	0,999	0,145	1,33	0,16	...
Segment3	1.0	2,41	0	2,0	0,178	...
Segment4	1,3	0,5	0	0,20	1,0	...
...	...	...	...	...	...	...

Figure 5.8 : V_1, V_2 et V_3 vecteurs des termes à classifiés.

A la fin du processus de classification, nous obtenons un résultat sous forme de classes de termes qui sont mutuellement disjointes c'est-à-dire en prenant deux classes quelconques de ce résultat leur intersection est vide (figure 5.9).

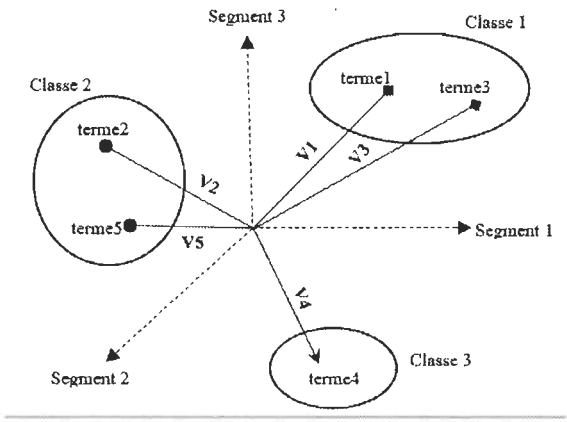


Figure 5.9 : résultat de classification des termes.

Algorithme :

Voici l'algorithme permettant la classification des termes :

Input : segments-termes, classifieur
Output : classe_du_termes
T : table de vecteur

Début

Pour chaque colonne de la matrice TF-IDF **Faire**

Vecteur ← une colonne de la matrice

Ajouter le vecteur à T.

Fin pour

Classe_du_termes ← classifier (classifieur, T)

Fin

Algorithme 5.6 : algorithme de classification des termes

5.4.3. Extraction des règles d'association :

La figure 5.10 montre le processus d'extraction des règles d'association dont il se décompose en trois étapes à savoir :

1. La fragmentation verticale.

2. L'extraction de sous- ensemble des règles d'association et

3. L'union des règles d'association.

La figure 5.10 montre le processus d'extraction des règles d'association. La première étape consiste à fragmenter la table de transaction en sous-tables (fragments) disjointes. L'ensemble des termes de chaque fragment appartiennent à la même classe de termes. Le nombre de fragments résultants dépend fortement de résultat de classification, autrement dit, la taille d'ensemble de fragments égale à celle d'ensemble de classe de termes. À la deuxième étape et par l'application de l'algorithme Apriori, nous extrayons les règles d'association de chaque fragment. Enfin, nous faisons l'union entre ces ensembles de règles afin de générer le résultat final.

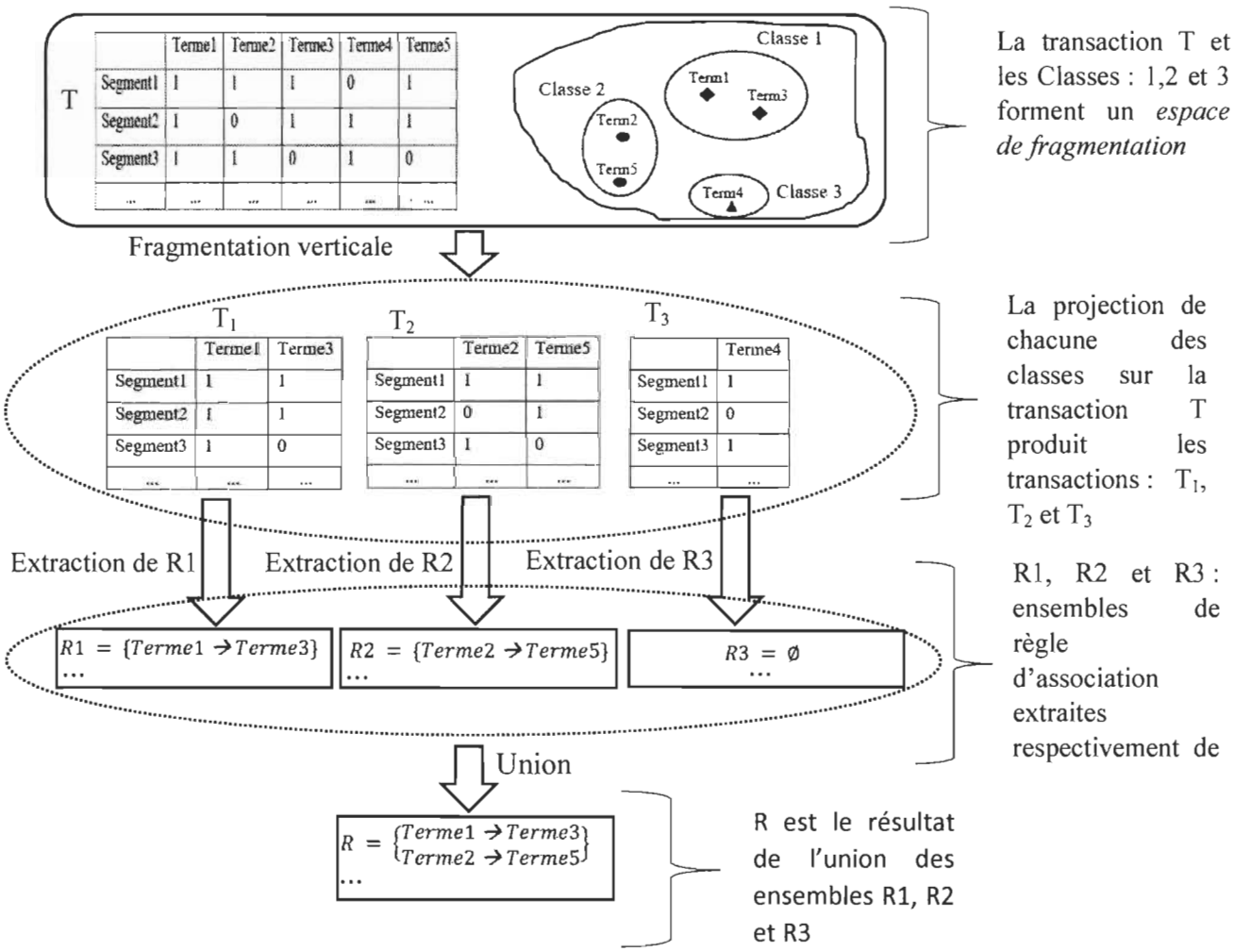


Figure 5.10 : processus d'extraction des règles d'association

5.4.3.1 La fragmentation verticale :

La fragmentation verticale d'une table de transactions (T) permettant de sélectionner les colonnes appartenant à chaque fragment. L'opération de fragmentation est obtenue grâce à la sélection des colonnes d'une transaction en l'occurrence les termes. Ces derniers appartenant à une classe bien précise (figure 5.11).

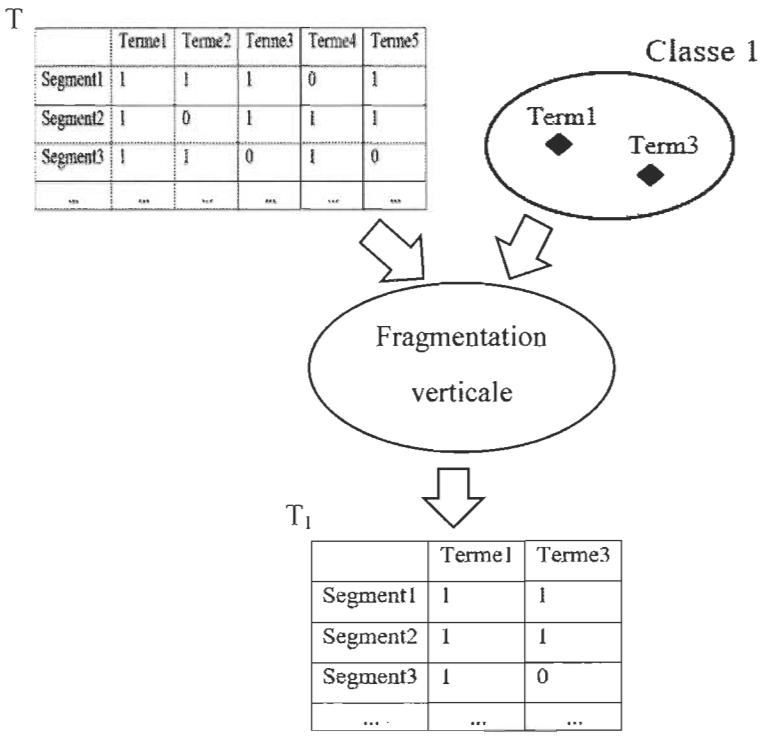


Figure 5.11 : fragmentation verticale en utilisant la Classe 1.

La table de transaction (T) est déduite de la matrice TF-IDF en utilisant cette procédure : Nous remplaçons chaque valeur non nul de la matrice TF-IDF par la valeur 1 sinon par la valeur 0 (figure 5.12).

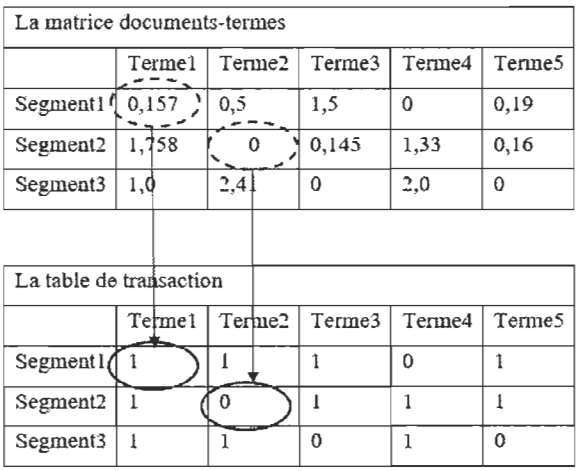


Figure 5.12 : déduction de la table de transaction

Algorithme :

Voici l'algorithme permettant la fragmentation d'une table de transaction :

Input : matrice segments_termes, résultat de classification

Output : fragments

Début
// Construire la table de transaction T
Pour chaque ligne dans segments_termes **faire**
    Pour chaque colonne dans segments_termes **faire**
        Si valeur > 0 **alors**
            T [ligne,colonne] ← 1
        Sinon
            T [ligne,colonne] ← 0
        Fin si
    Fin pour
Fin pour
// La fragmentation du la table de transaction T
Fragments ← ∅
Pour chaque classe C dans le résultat de classification **faire**
    Fragment' ← ∅
    Pour chaque colonne de T ∈ C **faire**
        Fragment' ← Fragment' ∪ colonne sélectionnée
    Fin pour
    Fragments ← Fragments ∪ Fragment'
Fin pour
Fin Tant que
Fin

Algorithme 5.7 : algorithme de fragmentation de table de transaction

5.4.3.2 L'extraction des règles d'association :

Pour chacun des fragments obtenus lors de l'étape de la fragmentation verticale nous considérons une sous-table de transaction. Par la suite, nous appliquons l'algorithme Apriori, paramétré au préalable par l'utilisateur, sur ces sous-tables afin d'extraire les sous-ensembles des règles d'association. Ces dernières sont mutuellement disjointes parce que les classes des termes y sont déjà.

Après avoir extrait les sous-ensembles des règles d'association, nous construisons le résultat final en appliquant l'opération d'union sur ces sous-ensembles.

Algorithme :

Voici l'algorithme permettant la production des règles d'association :

Input : Fragments, Apriori, paramètres
Output : sous-ensembles des règles d'association
Début
Ensemble ← ∅
    Pour fragment ∈ Fragments **faire**
        Ensemble ← Ensemble ∪ Apriori (fragment, paramètres)
    Fin pour
Fin

Algorithme 5.8 : algorithme pour la production des règles d'association

5.5. Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes de conception de notre méthode de sélection des mots clés. Le chapitre suivant mettra en lumière la partie expérimentale de notre projet. Par la suite, les résultats obtenus seront analysés et interprétés. Les résultats d'expérimentation sur des données réelles montrent que notre méthode de sélection de mot clés permet d'obtenir un ensemble de mot clé petit et satisfaisant.

Chapitre 6

Expérimentations et résultats

Dans ce chapitre, nous exposons les différents résultats obtenus lors des phases d'expérimentations. Les expérimentations sont effectués sur un livre complet à savoir : « La civilisation des Arabes »[49].

6.1 Architecture du système :

Dans notre système nous avons développé seize fonctions réparties en trois grands modules à savoir : classification, Apriori_algorithme et analyseur_texte. Dans ce qui suit nous détaillons le contenu de chaque module.

- **Le module classification :**

Le module classification permet principalement l'implémentation des méthodes de classification. Il est composé de neuf fonctions (figure 6.1). Les fonctions : «euclidean», «minkonshi» et «manhathan» implémentent la distance euclidienne, la distance de Minkonshi et la distance de Manhathan respectivement et les fonctions : «KNN», «SMO», «SVM», «ART1» implémentent les méthodes de classification. La fonction «afficher_classe» permet d'afficher le contenu d'une classe sélectionnée. Enfin, la fonction «rules» représente les règles d'association sous une format compréhensible.

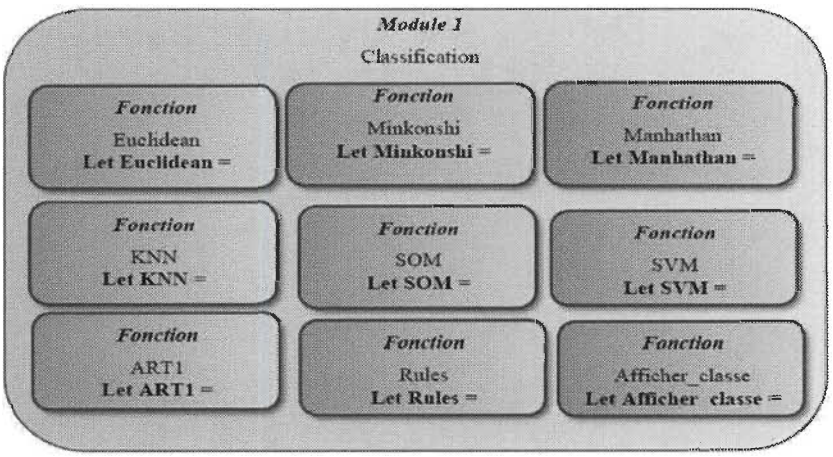


Figure 6.1 : le module classification.

- **Le module Apriori_algorithme :**

Le module Apriori_algorithme est composé d'une seule fonction dont le rôle est la mise en œuvre de l'algorithme Apriori (figure 6.2).

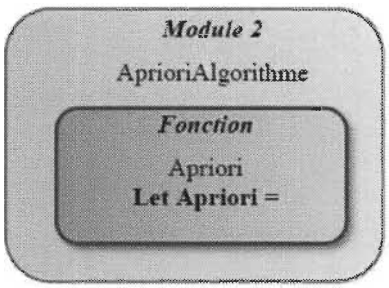


Figure 6.2 : le module Apriori_algorithme

- **Le module analyseur_texte :**

Ce module permet de construire un index inversé, l'initialisation de l'analyseur et la construction de la matrice tf-idf. Il est composé de six fonctions (figure 6.3). Ces fonctions sont :

- SetIndexWriter : permet l'initialisation d'un nouvel index en mode d'écriture.
- SetIndexReader : permet l'initialisation d'un index existant en mode de lecture.
- AjouterDocIndex : ajouter des segments à l'index.
- SetAnalyzer : initialise l'analyseur.
- TermIdMap : permet la construction d'une table de hachage ou chaque élément est représenté par <mot, id>.
- TdMatrix : la construction de la matrice tf-idf.

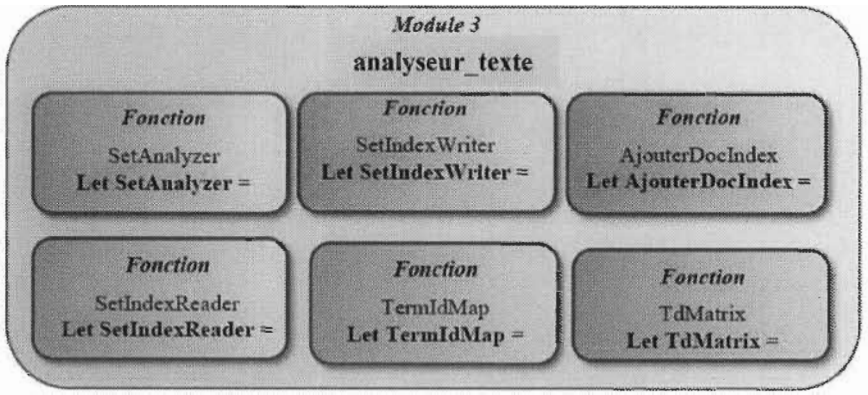


Figure 6.3 : le module analyseur_texte

6.2 Implémentation :

6.2.1 Langages choisis pour l'implémentation :

Nous avons développé notre système en utilisant la plateforme Windows et avec le logiciel Microsoft Studio 2015. Nous avons utilisé le langage F# de programmation fortement typé et multi-paradigme qui englobe des techniques de programmation fonctionnelle, impérative et orientée objet [50].

- **Langage fortement typé :** Les types de données utilisées représentent adéquatement les données manipulées [51].
- **Programmation fonctionnelle :** La programmation fonctionnelle est un paradigme de programmation dont le calcul est considéré comme étant une évaluation d'une fonction mathématique [52].
- **Programmation impérative :** La programmation impérative est un paradigme de programmation dont l'état du programme est modifié par des séquences d'opérations sous forme d'instruction [53].
- **Programmation orientée objet :** La programmation orientée objet modélise des objets ayant un état (ensemble de variables) et des méthodes (fonctions) qui leur sont propres.

6.2.2 Les interfaces :

Au lancement de l'exécution de l'application une fenêtre est apparue (figure 6.4). Les composants principaux de cette interface sont :

- Ajouter fichiers : permet de charger des documents à analyser situés dans n'importe quel répertoire de l'ordinateur.
- Options : permet d'afficher les paramètres de classifieur, l'algorithme Apriori et le type de segmentation dans une fenêtre pour les confirmer ou les modifier (figure 6.5).
- Tous supprimer : supprimer tous les fichiers déjà choisis.
- Supprimer : supprimer un fichier sélectionné.
- Analyser : lancer la fenêtre d'analyse du texte.

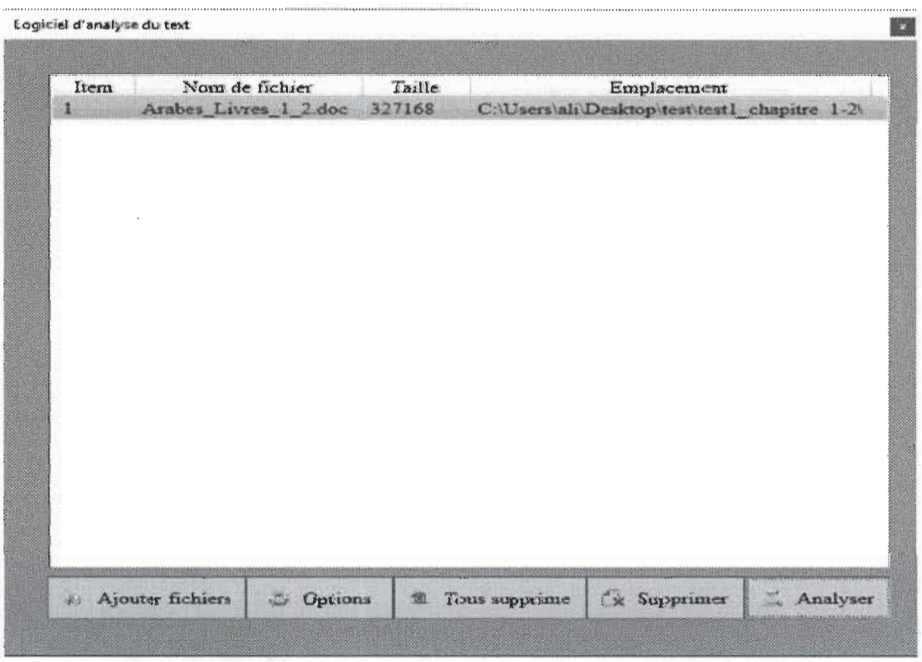


Figure 6.4 : l'interface principale

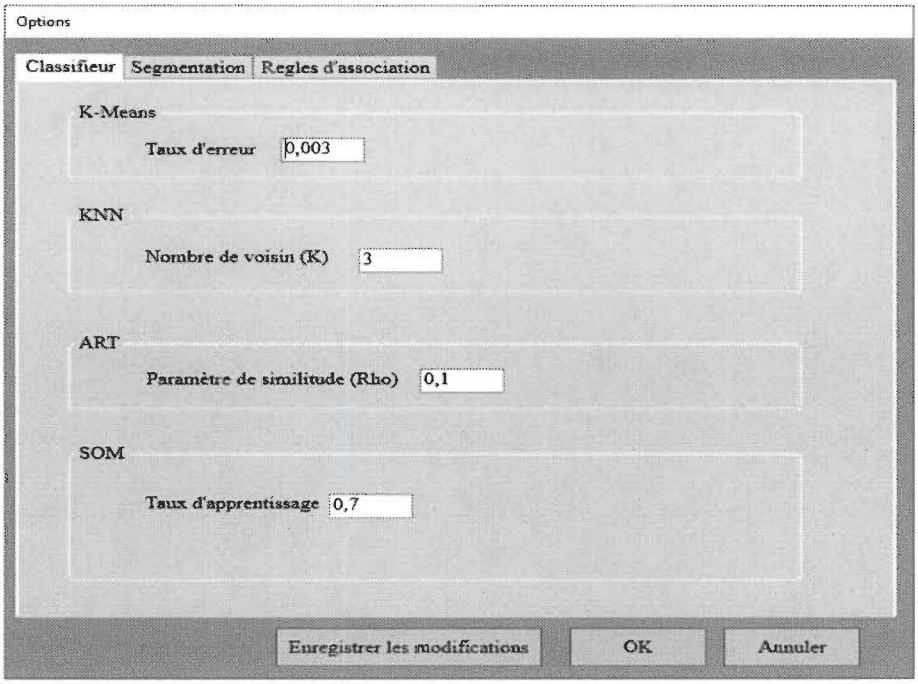


Figure 6.5 : fenêtre d'options.

Après le choix du buton «Analyser» une fenetre apparait (figure 6.6). Elle est divisée en deux zones comme suit :

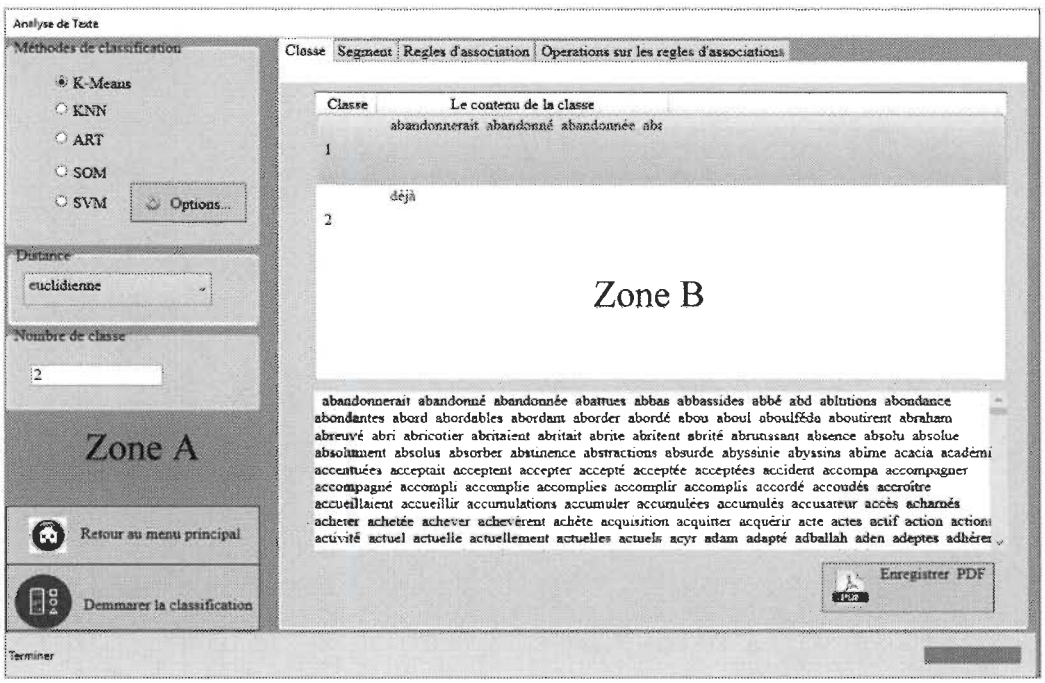


Figure 6.6 : la fenetre d'analyse du texte.

La zone « A » concerne le choix de la méthode de classificaction, le type de distance et le paramètre «nombre de classes». Le bouton «Retour au menu principal» permet de revenir à la fenêtre principale et le bouton «Demarrer la classification» demarre l'exécution.

La zone « B » concerne l'affichage textuel des résultats obtenus. Elle permet à l'utilisateur de visualiser et de sauvegarder en pdf le contenu des classes, des segments et des règles d'association tout en se déplaçant entre les onglets : «classe» , «segment» et «règles d'association».

Après la selection de l'onglet «Operation sur les règles d'association» un affichage apparait (figure 6.7). Cet onglet contient l'opération d'intersection et les zones d'affichage dans lesqueles le détail du résulat s'affiche. Le bouton «Intersection» permet de lancer l'opération d'intersection sur les règles sélectionnées et le bouton «Enregistrer PDF» enregistre le résultat dans le répertoire courant de l'application.

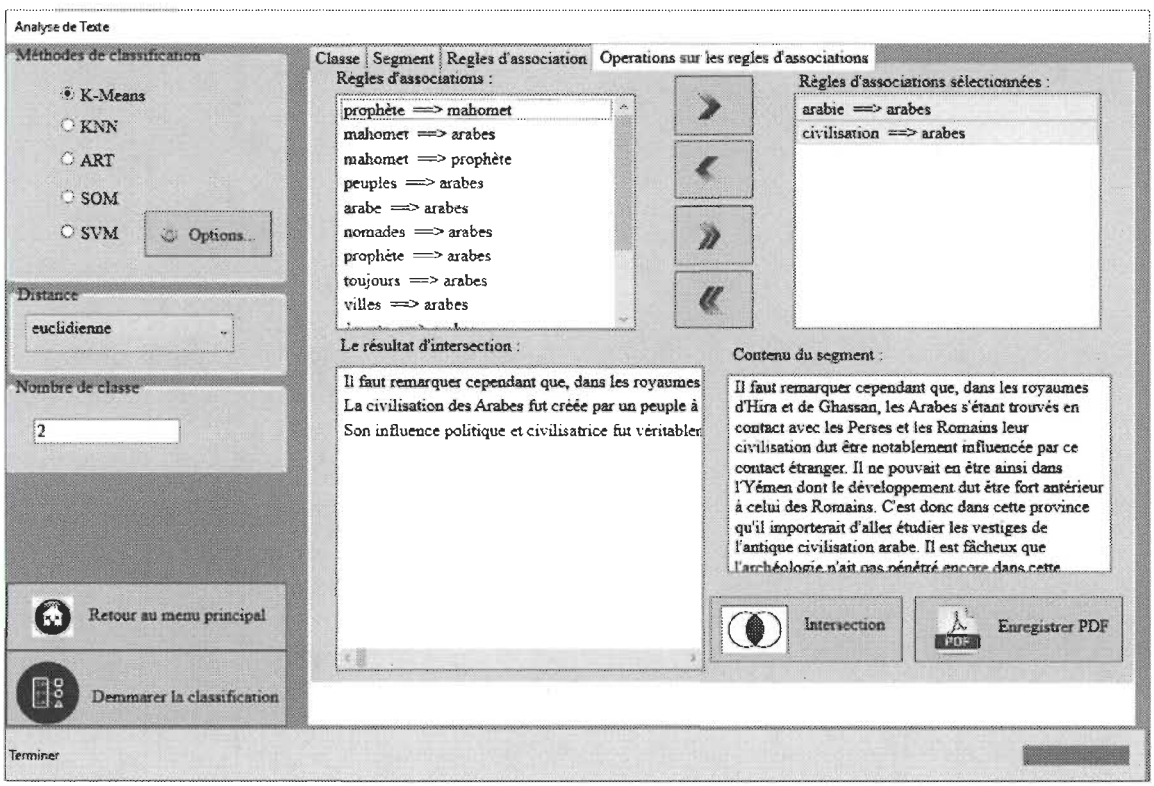


Figure 6.7 : onglet d'operation sur les règles d'association.

6.3 Expérimentations : « La civilisation des Arabes » :

Le livre est déjà subdivisé en six chapitres regroupés en quatre parties : « chapitre 1 & chapitre 2 », « chapitre 3 », « chapitre 4 » et « chapitre 5 & chapitre 6 ». Dans cette expérimentation, nous gardons le même regroupement afin de réduire le temps de traitement.

Pour chaque partie, on définira les paramètres suivants qui serviront pour toute la phase d'expérimentation :

- Suppression de la table des matières et la liste des figures.

- Suppression des tableaux contenant des références aux figures du livre en édition papier de 1980 pages.

- Suppression de l'hyper lien « Retour à la table des figures ».

- Selon quelques essais, on constate que le choix du classifieur n'affecte pas l'ensemble des mots clés. À cet égard, on applique l'algorithme K-mean avec un taux d'erreur 0,003 parce qu'il est rapide en termes de temps d'exécution. Le nombre de classes affecte le degré de parallélisme qui indique combien de processus d'extraction de règles d'association sont simultanément actifs. Dans cette expérimentation, le nombre de classes égale deux.

- Le choix de type de segmentation affecte l'ensemble des mots clés. Dans le cadre de ces expérimentations, c'est la segmentation en paragraphe qui a été choisie. Ceci est dû au fait que le choix des autres types génère un ensemble de mots clés vides.

- Selon quelques tests, on choisit la valeur du support et la valeur de la confiance. Si on choisit de petites valeurs, celles-ci peuvent conduire à un ensemble de mots clés de grande taille. Tandis que, avec des grandes valeurs l'ensemble de mots clés peut être vide ou bien ne sera pas suffisant pour refléter les mots du corpus. Pour les chapitres « chapitre 4 » et « chapitre 1 & chapitre 2 », la valeur du support est 3% et le reste des

parties est 5%. La valeur de la confiance est la même pour tous les chapitres qui égale à : 30%.

6.3.1 Partie 1 :

Les livres de cette partie ainsi et leur table des matières correspondante sont :

Livre premier : Le milieu et la race

Chapitre 1 : L'Arabie

Chapitre II : Les Arabes

Chapitre III : Les Arabes avant Mahomet

Livre deuxième : Les origines de la civilisation arabe

Chapitre I : Mahomet. Naissance de l'empire arabe.

Chapitre II : Le Coran

Chapitre III : Les conquêtes des Arabes.

Dans cette analyse, 692 segments ont été obtenus et deux classes de termes. Ci-après on montre un extrait de ces résultats.

Segments	Contenu
Segment :273	Arabes de la Chine. - Aussitôt que l'empire des Arabes fut constitué, les khalifes d'Orient et les souverains de la Chine s'envoyèrent fréquemment des ambassadeurs et comme nous le verrons dans une autre partie de cet ouvrage, les relations commerciales des Arabes avec la Chine s'établirent régulièrement par voies de terre et de mer.
Segment :247	Arabes de la Syrie. - Comme les Arabes de l'Arabie, ceux de la Syrie se distinguent en nomades et sédentaires. Les premiers habitent le désert ; les seconds, les villes.
Segment :637	C'est avec Omar que commence en réalité l'empire des Arabes. Obligé de quitter la Syrie et de se réfugier dans Constantinople, sa capitale, l'empereur Héraclius comprit que le monde allait avoir de nouveaux maîtres.
Segment : 20	La civilisation des Arabes règne depuis douze siècles sur l'immense région qui s'étend des rivages de l'Atlantique à la mer des Indes, des

	plages de la Méditerranée aux sables de l'Afrique intérieure. Les populations qui l'habitent possèdent la même religion, la même langue, les mêmes institutions, les mêmes arts, et firent jadis partie du même empire.
Segment :263	En dehors des Arabes, la population de l'Égypte contient des éléments fort divers : Turcs, Coptes, Syriens, Nègres, Grecs et Européens, etc. ; mais les croisements de ces éléments divers avec le fellah sont fort rares. Du reste, le climat de l'Égypte est tellement meurtrier pour l'étranger, qu'on ne cite guère d'individu de nationalité étrangère, y compris les Turcs, qui ait jamais pu se reproduire au-delà de la deuxième génération. L'Arabe est le seul peuple étranger qui ait réussi à faire souche en Égypte.
Segment :326	Les auteurs bibliques nous parlent souvent du commerce des Arabes, des villes qu'ils possédaient, et notamment de Saba, dans l'Yémen ; mais si les indications qu'ils nous donnent révèlent l'existence de grandes villes à une époque fort reculée, ils ne nous fournissent aucun document sur elles.
Segment :417	Mahomet finit, du reste, par comprendre qu'il n'était pas toujours avantageux d'avoir trop de femmes à la fois, car il défendit à ses disciples d'en avoir plus de quatre en même temps. Il n'est pas besoins d'ajouter que ce n'est pas lui qui établit la polygamie chez les Arabes ; elle existait bien avant le prophète chez tous les peuples de l'Asie, quel que fût leur culte, et existe encore.
Segment :406	Nous nous sommes surtout occupé dans ce qui précède de la vie publique de Mahomet. Il nous reste maintenant à essayer de reconstituer le caractère et la vie privée du prophète, d'après les documents que les Arabes nous ont laissés.
Segment :101	Le désert est parcouru constamment par des tribus nomades. La vie au désert, qui semblerait si affreuse à un Européen, est tellement pleine de charme pour les nomades qu'ils la préfèrent à toute autre, et leur préférence ne date pas d'hier, car les nomades d'aujourd'hui sont fils des Arabes dont nous parle la Bible. Ils en ont conservé les goûts, les moeurs et les coutumes.
Segment :245	Arabes de l'Arabie. - L'Arabe des régions centrales de l'Arabie est celui qui, malgré ses mélanges répétés avec des Nègres, semble être resté le plus semblable à ses ancêtres des premiers âges, surtout si on le considère à l'état nomade. Ce sont donc ces derniers que nous étudierons d'abord.

Classes	Contenu
Classe : 1	…**arabes arabie** argent arts asie assez atlantique aucun aucune aujourd autrefois autres beaucoup… **civilisation** communs compagnons complexes comptaient diverses djôf documents durent déjà désert **empire** empêche enfants entièrement …ismaélite isolées jamais jugent jugeons jusqu kahtanite katif langue lecteur lesquels littoral lois longtemps lutter **mahomet** malgré mecque milieu milieux monde montagnes mulâtres mères mélange mélanges mélangé mélangée mélangés nations nedjed noir noire noirs **nomades** nombre nombreux nouvelle nouvelles peuple **peuples** peuvent phéniciens plusieurs poignée pourrait pouvait pouvons premier premiers presque produit **prophète** proportions prouvent préjugé…**siècles** souvent suite suivant surtout **syrie** sédentaires teintes temps tion **toujours** transformer tribus trouve trouvé turcomans turcs type types variés viennent ville **villes** visage visité voyage voyageurs wallin yémen âges également **égypte** égyptiens élément éléments époque épée étude
Classe : 2	abandonnerait abandonné abattues abbas abbassides abbé abd ablutions abondance abondantes abord abordables abordant aborder aboul aboulféda aboutirent abraham abreuvé abri abricotier abritait abrite abritent abrité abrutissant absence absolu absolus absorber abstinence abstractions absurde abyssinie abyssins abîme acacia académie accentuées acceptait acceptent accepter accepté acceptée …

Le tableau ci-après est un extrait d'une transaction (T1) qui représente le résultat de la fragmentation verticale sur la transaction globale en utilisant les termes de la première classe. Tous les termes du tableau appartiennent à la première classe.

	arabes	civilisation	prophète	mahomet	peuples	arabie	nomades	toujours	villes	égypte	siècles	syrie	empire	…
S : 273	1	0	0	0	0	0	0	0	0	0	0	0	1	…
S : 247	1	0	0	0	0	0	0	1	0	0	1	1	0	…
S : 637	1	0	0	0	0	0	0	0	0	0	0	0	0	…
S : 20	1	1	0	0	0	0	0	0	0	0	1	0	1	…
S : 263	1	0	0	0	0	0	0	0	0	1	0	0	0	…
S : 326	1	0	0	0	0	0	0	0	1	0	0	0	0	…
S : 417	1	0	1	1	1	0	0	1	0	0	0	0	0	…
S : 406	1	0	1	1	0	0	0	0	0	0	0	0	0	…
S : 101	1	0	0	0	0	0	1	0	0	0	0	0	0	…
S : 245	1	0	0	0	0	1	1	0	1	0	0	1	0	…

L'extraction des règles d'association sur les données précédentes conduit aux résultats présentés dans le tableau 6.1. On remarque que tous les membres de chaque règle appartiennent à la classe une et les données de la deuxième classe ne génèrent aucune règle. Rappelons que la valeur du support égale 3% et la valeur de la confiance égale 30%.

Règle d'association	**Support**	**Confiance**
civilisation → arabes	6%	70%
prophète → Mahomet	6%	70%
mahomet →arabes	6%	30%
mahomet → prophète	6%	30%
peuples → arabes	5%	64%
arabie –> arabes	5%	57%
nomades → arabes	4%	71%
prophète → arabes	3%	40%
toujours → arabes	3%	39%
villes → arabes	3%	62%
égypte → arabes	3%	65%
siècles → arabes	3%	67%
syrie → arabes	3%	60%
empire → arabes	3%	47%

Tableau 6.1 : résultats d'expérimentation de la partie 1

Les résultats du tableau 6.1 montrent une relation du « civilisation » et le mot « arabes » avec une confiance de 70%. On constate aussi d'autres relations entre les mots « Mahomet », « peuples », « prophète », « nomades », « empire », et le mot « arabes » avec respectivement les confiances : 30%, 64%, 40%, 71% et 47%.

On aurait pu mettre la valeur de confiance a 10%. Avec cette valeur d'autres règles s'ajoutent à l'ensemble de règles du tableau 6.1. Le tableau 6.2 montre les nouvelles règles obtenues. On remarque que tous les mots jouent le rôle d'antécédent et de conséquent. Par conséquent, la propriété d'apparition seulement au membre droit des règles dont elle caractérise le mot « arabes » dans les règles du tableau 6.1 a disparu.

Règle d'association	**Support**	**Confiance**
arabes → civilisation	6%	23%
Mahomet → prophète	6%	30%
Arabes → mahomet	6%	21%
Prophète → mahomet	6%	30%
arabes → peuples	5%	18%
Arabes → arabie	5%	18%
Arabes → nomades	4%	15%
arabes → prophète	3%	12%
Arabes → toujours	3%	11%
Arabes → villes	3%	11%
Arabes → égypte	3%	11%
Arabes →siècles	3%	10,5%
arabes → syrie	3%	10,5%
arabes → empire	3%	10,5%

Tableau 6.2 : nouvelles règles si la confiance = 10 %

Un mot est dit attribut de classe ou bien cible si et seulement si pour toutes les règles d'association résultantes il se trouve au membre droit de chacun d'eux. Ainsi, le mot « Arabes » est un attribut de classe car il appartient seulement à la partie droite de toutes les règles d'association résultantes d'où les règles suivantes :

civilisation → arabes

Mahomet →arabes

peuples → arabes

Arabie –> arabes

nomades → arabes

prophète → arabes

toujours → arabes

villes → arabes

Égypte → arabes

siècle → arabes

Syrie → arabes

empire → arabes

sont considérées comme des règles de classification. Les règles d'association sont dites de classification si est seulement si elles sont limitées en leur côté droit par l'attribut de classe. On remarque que le mot « Mahomet » n'est pas un attribut de classe à cause de la règle : « Mahomet → prophète », idem pour le mot « prophète ».

Ainsi, l'ensemble des règles de classification et l'ensemble des attributs de classe du tableau 6.1 sont :

- Règles de classification =

{civilisation → arabes

Mahomet →arabes

peuples → arabes

Arabie –> arabes

nomades → arabes

prophète → arabes

toujours → arabes

villes → arabes

Égypte → arabes

siècle → arabes

Syrie → arabes

empire → arabes}

- Attributs de classe = {arabes}

L'ensemble de mots clés sélectionnés sont {civilisation, Mahomet, peuples, Arabie, nomades, prophète, toujours, villes, Égypte, siècle, Syrie, empire}. Ces mots sont étroitement corrélés avec l'attribut de classe en l'occurrence « arabes ». Cet ensemble est cohérent avec les chapitres. En effet, les chapitres démontrent la civilisation des arabes et leur origine et un énorme empire qui durée plusieurs siècles et s'étendit sur plusieurs villes.

6.3.2 Partie 2 :

Le livre de cette partie et sa table des matières correspondante sont :

Livre troisième : L'empire des Arabes

Chapitre I : Les Arabes en Syrie

Chapitre II : Les Arabes à Bagdad

Chapitre III : Les Arabes en Perse et dans L'Inde

Chapitre IV : Les Arabes en Égypte

Chapitre V : Les Arabes dans l'Afrique septentrionale

Chapitre VI : Les Arabes en Espagne

Chapitre VII : Les Arabes en Sicile, en Italie et en France

Chapitre VIII : Luttes du christianisme contre l'islamisme. Les croisades.

Dans cette analyse, 761 segments ont été obtenus et deux classes de termes. Ci-après on montre un extrait de ces résultats.

Segments	**Contenu**
Segment : 241	Ensanglantée chaque jour par les dissensions religieuses, ruinée par les exactions des gouverneurs, l'Égypte professait une haine profonde pour ses tristes maîtres, et devait recevoir comme libérateurs ceux qui l'arracheraient aux mains des empereurs de Constantinople. C'est aux Arabes que fut réservé ce rôle.
Segment : 418	La conquête de l'Afrique par les Arabes fut beaucoup plus difficile que celle de l'Égypte, et ils ne s'y établirent que très lentement. Les Berbères ne cessèrent de lutter contre eux, et, à plusieurs reprises, arrivèrent à reconquérir leur indépendance.
Segment : 108	La grande mosquée de Damas est construite sur le même plan que les premières constructions analogues de l'islamisme et se compose comme elles d'une grande cour rectangulaire à portiques, dont un côté est occupé par le sanctuaire et les angles par les minarets. Dans le chapitre consacré aux Arabes en Égypte nous aurons à décrire plusieurs monuments du même type.
Segment : 551	Monuments arabes de Séville. - Séville est, comme Tolède, bien qu'à un point de vue un peu différent, une cité où l'influence arabe se retrouve à chaque pas. L'architecture de la plupart des maisons modernes est arabe ; les danses et la musique populaire également arabes. L'influence du sang arabe y est reconnaissable chez les femmes, surtout, à bien des détails.
Segment : 165	Pour un petit nombre de contrées, la Perse notamment, les renseignements que nous possédons sont rares, et nous serons obligés de nous contenter d'indications sommaires. Elles suffisent cependant à prouver que l'influence que les Arabes y ont exercée, de même du reste que celle qu'ils y ont subie, a été très grande.
Segment : 608	Ces dissensions seules rendirent possible la conquête définitive de la Sicile. Elle fut terminée en 1072 par la prise de Palerme. C'est de cette époque que l'on peut dire que la puissance politique des Arabes en Sicile disparut ; mais grâce à la sagesse de Roger et de ses successeurs, leur influence civilisatrice dura longtemps.

Segment : 419	Après avoir été soumise aux Romains durant plusieurs siècles, l'Afrique septentrionale avait été dominée pendant plus de cent ans (429-545) par les Vandales d'Espagne. Ils en furent chassés par l'expédition envoyée contre eux par Justinien et dirigée par Bélisaire. Les Visigoths d'Espagne l'envahirent à leur tour et l'occupaient en partie quand les Arabes se présentèrent.
Segment : 227	L'étude des oeuvres plastiques des Arabes en Égypte prouvera combien cette substitution a été complète. Bien que le pays fût couvert de nombreux monuments anciens, les Arabes ne leur ont rien emprunté.
Segment : 215	Palais du grand Mongol, à Delhi, ou Fort de Shah Jehan. - Ce palais, construit par Shah Jehan, fut terminé en 1058 de l'hégire (1640 de J.-C.). Il passait pour le plus beau palais musulman existant dans l'Inde et dans la Perse. Les mosaïques des salles faisaient de chacune d'elles une véritable pièce d'orfèvrerie.

Classes	**Contenu**
Classe : 1	**....arabes** arabesque arabie aragon arbitre arbre arc arca arcade architecte escalier escaliers **civilisation** esclaves escortent escurial espa **Espagne** ibn ibère ibères identique identiques idiomes idole idée ifrikia infligé **influence** infor informa infructueux louxor loyauté lucain lueur lumineux... montagne mérovingiens mésopotamie métal métans métaux églises égorgent **égypte** ...égyptiens élancer élargie élection élective oeuvres
Classe : 2	arabe arabisée berbère langue.

L'extraction des règles d'association à partir de données précédentes mène aux résultats présentés dans le tableau 6.3. On constate que tous les membres des règles (antécédents, conséquents) appartiennent à la classe une et les données de deuxième classe ne génèrent aucune règle. Rappelons que la valeur du support égale 5% et la valeur de la confiance égale 30%.

Le tableau suivant est un extrait d'une transaction (T1) qui représente le résultat de la fragmentation verticale sur la transaction globale en utilisant les termes de la première classe. Tous les termes du tableau appartiennent à la première classe.

...	S : 215	S : 227	S : 419	S : 608	S : 165	S : 551	S : 108	S : 418	S : 241	
...	0	0	0	1	1	1	0	0	0	**influence**
...	0	0	0	0	0	0	0	0	0	**civilisation**
...	0	0	1	0	0	0	0	0	0	**espagne**
...	0	1	0	0	0	1	1	0	0	**monuments**
...	0	1	0	0	0	0	1	1	1	**egypte**
...	...	...	...	...	...	...	...	...	...	**...**

Règle d'association	Support	Confiance
Égypte → Arabes	5,5%	46%
influence → Arabes	5,65%	70,50%
Espagne → Arabes	6,83%	65%
monuments → Arabes	7%	58,69%
civilisation → Arabes	8,40%	80%

Tableau 6.3 : résultats d'expérimentation de la partie 2

Les résultats du tableau 6.3 démontrent une forte relation entre le mot « civilisation » et le mot « arabes » avec une confiance de 80%. On constate aussi d'autres relations entre le mot « influence » et le mot « arabes » avec une confiance de 70,50%. On remarque que la relation entre le mot « Espagne » et le mot « arabes » a une confiance de 31%.

On aurait pu mettre la valeur de confiance a 10%. Avec cette valeur d'autres règles s'ajoutent à l'ensemble de règles du tableau 6.3. Le tableau 6.4 montre les nouvelles règles obtenues. On remarque que tous les mots jouent le rôle d'antécédent et de conséquent. Par conséquent, la propriété d'apparition seulement au membre droit des règles dont elle caractérise le mot « arabes » dans les règles du tableau 6.3 a disparu.

Règle d'association	Support	Confiance
Arabes → Égypte	5,5%	14%
Arabes → influence	5,65%	14,5%
Arabes → Espagne	6,83%	17,5%
Arabes → monuments	7%	18,24%
Arabes → civilisation	8,40%	21%

Tableau 6.4 : nouvelles règles si la confiance =10%.

L'attribut de classe fait partie du membre droit de tous les règles d'association résultantes. Ainsi, Le mot « arabes » est un attribut de classe car il appartient seulement à la partie droite de toutes les règles d'association résultantes d'où les règles suivantes :

Égypte → arabes

influence → arabes

Espagne → arabes

monuments → arabes

civilisation → arabes

sont considérées comme des règles de classification. Ces derniers sont toutes les règles avec l'attribut de classe dans leur côté droit [54-56].

Ainsi, l'ensemble des règles de classification et l'ensemble des attributs de classe du tableau 6.3 sont :

- Règles de classification =

{Égypte → arabes

influence → arabes

Espagne → arabes

monuments → arabes

civilisation → arabes}

- Attributs de classe = {arabes}.

L'ensemble des mots clés est {Égypte, influence, Espagne, monuments, civilisation}. Ces mots sont étroitement corrélés avec l'attribut de classe en l'occurrence « arabes ». Cet ensemble est cohérent avec les chapitres d'expérimentation. En effet, les chapitres démontrent l'influence de la civilisation des arabes en Égypte et en Espagne.

6.3.3 Partie 3 :

Le livre de cette partie et sa table des matières correspondante sont :

Livre quatrième : Les mœurs et les institutions des Arabes.

Chapitre I : Les Arabes nomades et Arabes sédentaires des campagnes.

Chapitre II : Les Arabes des villes. - Mœurs et coutumes.

Chapitre III : Institutions politiques et sociales des Arabes

Chapitre IV : Les femmes en Orient.

Chapitre V : Religion et morale.

Dans cette analyse, 387 segments ont été obtenus et deux classes de temres.ci-après on montre un extrait de ces résultats.

Segments	Contenu
Segment : 203	Le Coran est entré dans peu de développements sur le droit de propriété, mais tout ce qui le concerne a été bien réglé par les commentateurs. Ce droit a toujours été très respecté par les Arabes, même à l'égard des peuples vaincus. La terre, qui était enlevée à ces derniers par la conquête, leur était rendue moyennant un tribut qui dépassait rarement le cinquième de la récolte.
Segment : 229	Ces révoltes continuelles des gouverneurs étaient une cause très grande de faiblesse pour les khalifes ; mais la constitution politique de l'empire arabe en comportait bien d'autres. Une des plus profondes fut la diversité des races où régnait le Coran depuis le Maroc jusqu'à l'Inde. Très bien adapté aux besoins de certains peuples, le Coran ne l'était pas également aux besoins de tous.
Segment : 23	Les traits principaux de cette esquisse seront puisés dans l'observation des Arabes actuels. Une telle méthode n'est applicable qu'à un petit nombre de peuples. Il est facile de montrer qu'elle l'est surtout aux populations de l'Orient dont nous étudions l'histoire.
Segment : 179	Cette étude préalable des peuples dont on veut décrire et surtout comprendre l'organisation sociale, est indispensable, qu'il s'agisse des Arabes ou d'un peuple quelconque. Il est à souhaiter que les juristes finissent un jour par en comprendre l'importance. La science du droit cessera alors d'être constituée par de sèches énumérations d'articles de lois compliquées de dissertations véritablement

	byzantines.
Segment : 269	L'islamisme a relevé la condition de la femme, et nous pouvons ajouter que c'est la première religion qui l'ait relevée. Il est facile de le prouver en montrant combien la femme a été maltraitée par toutes les religions et tous les peuples qui ont précédé les Arabes. Nous nous sommes déjà expliqués sur ce point dans notre dernier ouvrage et n'avons qu'à répéter ce que nous y avons dit pour convaincre le lecteur.
Segment : 61	Ainsi qu'il arrive dans toutes les régions voisines du désert, c'est-à-dire par conséquent dans la plus grande partie de l'Arabie, les populations sédentaires du Haouran se trouvent en contact avec des Arabes nomades, qui, ne pouvant vivre uniquement des produits de leurs troupeaux et de l'élevage des chevaux et chameaux, sont obligés de s'adonner au pillage.
Segment : 284	L'autorité du père de famille, si faible aujourd'hui chez les peuples chrétiens, a conservé en Orient toute sa force. Les femmes ne parlent qu'avec le plus grand respect à leurs maris, et les enfants suivent naturellement cet exemple. Le père de famille possède en réalité toute l'autorité et les privilèges de celui de la Rome antique. Sur ce point encore, les Orientaux ne nous envient point.
Segment : 233	À quelque époque qu'aient régné les Arabes, ou les peuples divers qui continuèrent après eux à propager le Coran, leurs institutions politiques se sont toujours présentées sous forme de monarchie militaire et religieuse absolue. Puissantes pour fonder rapidement de grands empires,de telles institutions réussissent rarement à les faire durer. L'histoire des Arabes, des Mongols et des Turcs en donne la preuve. Ayant à lutter journellement contre les difficultés de toutes sortes au dedans et au dehors, ces grands empires n'ont de chance de prospérer que lorsqu'ils ont à leur tête des hommes tout à fait supérieurs.
Segment : 283	Les femmes sont entourées en Orient d'une surveillance sévère ; elles ne reçoivent jamais de visites d'hommes, et ne sortent que la figure voilée. Sauf peut-être à Constantinople, elles sont généralement accompagnées et ont bien rarement par conséquent occasion d'être tentées. Il ne faut donc pas trop nous étonner de voir les Orientaux soutenir que la vertu de leurs femmes est fort supérieure à celle des Européennes.

Classes	Contenu
Classe : 1	...**arabes...**bazar bazars beau **beaucoup**...**coran civilisation** fataliste fatalistes fatigante fatigue fatima faudrait fausse faut faute faux faveur faveurs favorable favori favoris favorisées façon faîte faïences fellah fellahines fellahs **femme femmes** fenêtres fer fera ferais ferait fermant fermer fermée mères mètres méchante mécomptes méconnaître médiaire médine médiocre médiocres méditerranée méthode méthodes métiers mêlée **mœurs**...nocturne noir noire noires noirs noix nom nomade **nomades** ...nombreuses nombreux nommé nommée nommées **toujours beaucoup** petits peuplades peuple peuplent **peuples...**
Classe : 2	acide fumée nicotine principes prussique recherches tabac toxique

L'extraction des règles d'association à partir de données précédentes mène aux résultats présentés dans le tableau 6.5. À la lumière des résultats précédentes on constate que tous les membres des règles (antécédents, conséquents) appartiennent à la classe une et les données de deuxième classe ne génèrent aucune règle. Rappelons que la valeur du support égale 3% et la valeur de la confiance égale 30%.

Le tableau suivant est un extrait d'une transaction (T1) qui représente le résultat de la fragmentation verticale sur la transaction globale en utilisant les termes de la première classe. Tous les termes du tableau appartiennent à la première classe.

	peuples	coran	arabes	nomades	toujours	beaucoup	nomades	mœurs	civilisation	institution	femme	orient	…
S : 203	1	1	1	0	1	0	0	0	0	0	0	0	…
S : 229	1	1	0	0	0	0	0	0	0	0	0	0	…
S : 23	1	0	1	0	0	0	0	0	0	0	1	0	…
S : 179	1	0	1	0	0	0	0	0	0	0	0	0	…
S : 269	1	0	1	0	0	0	0	0	0	1	0	0	…
S : 61	0	0	1	1	0	0	0	0	0	0	1	0	…
S : 284	1	0	0	0	0	0	0	0	0	0	1	0	…
S : 233	1	1	1	0	1	0	0	0	0	0	0	0	…
S : 283	0	0	0	0	0	0	0	0	0	0	1	0	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…

Règle d'association	Support	Confiance
peuples → coran	3%	32%
peuples → arabes	3%	32%
nomades → arabes	3%	85%
toujours → arabes	3%	40%
beaucoup → femmes	3%	41%
mœurs → arabes	3,87%	55,55%
civilisation → arabes	4,13%	66,66%
femme → femmes	4,39%	51,51%
coran → arabes	4,65%	38,29%
institutions → arabes	5,16%	55,55%
femmes → orient	5,42%	35,59%
orient → femmes	5,42%	38,88%

Tableau 6.5 : résultats d'expérimentation de la partie 3

Les résultats du tableau 6.5 démontrent une relation entre le mot « nomades » et le mot « arabes » avec une confiance de 85%. On constate aussi d'autres relations entre les mots « mœurs », « civilisation », « institutions » et le mot « arabes » avec respectivement les confiances : 55,55%, 66,66% et 55,55%.

On aurait pu mettre la valeur de confiance a 10%. Avec cette valeur d'autres règles s'ajoutent à l'ensemble de règles du tableau 6.5. Le tableau 6.6 montre les nouvelles règles obtenues. On remarque que tous les mots jouent le rôle d'antécédent et de conséquent. Par conséquent, la propriété d'apparition seulement au membre droit des règles dont elle caractérise le mot « arabes » dans les règles du tableau 6.5 a disparu.

Règle d'association	Support	Confiance
arabes → nomades	3%	11%
arabes → femme	3%	11%
orient → arabes	3,35%	24%
arabes → orient	3,35%	12%
femme → arabes	3%	20%
arabes → toujours	3,60%	12,80%
arabes → mœurs	3,80%	13,70%
arabes→ couran	4,65%	16,50%
femmes → femme	3,60%	23%
femmes→ beaucoup	3,60%	23%

Tableau 6.6 : nouvelles règles si confiance = 10%

La règle : femme → femmes, est une règle réflexive et n'ajoute aucune information pertinente. Elle n'est pas importante et représente, dans cet expérimentation, un bruit.

L'attribut de classe fait partie du membre droit de toutes les règles d'association résultantes. On remarque que le mot « Arabes » est un attribut de classe car il appartient seulement à la partie droite de toutes les règles d'association résultantes. Par conséquent les règles auxquelles il appartient : « peuples → arabes », « nomades → arabes », « mœurs → arabes », « civilisation → arabes », « coran → arabes » et « institutions → arabes » s'ajouteront à l'ensemble des règles de classification. Remarquons que le mot « coran » n'est pas un attribut de classe à cause de la règle : « coran → arabes », idem pour les mots « femmes » et « orient ».

Ainsi, l'ensemble des règles de classification et l'ensemble des attributs de classe du tableau 6.5 sont :

- Règles d'association avec contrainte = {

peuples → arabes

nomades → arabes

mœurs → arabes

civilisation → arabes

coran → arabes

institutions → arabes}

- Attributs de classe = {arabes}.

L'ensemble des mots clés est {peuples, nomades, mœurs, civilisation, coran, institutions}. Ces mots sont étroitement corrélés avec l'attribut de classe en l'occurrence « arabes ». Cet ensemble est un résultat cohérent avec les chapitres qu'on a utilisé pour notre expérimentation. Ils montrent entre autres les mœurs et institutions des arabes avant et après la révélation de Mahomet aux peuples arabes et la vie des nomades arabes.

6.3.4 Partie 4 :

Les livres de cette partie et leur table des matières correspondante sont :

Livre cinquième : La civilisation des Arabes.

Chapitre I : Origine des connaissances des Arabes. Leur enseignement et leurs méthodes.

Chapitre II : Langue, philosophie, littérature et histoire.

Chapitre III : Mathématiques et astronomie.

Chapitre IV : Sciences géographiques.

Chapitre V : Sciences physiques et leurs applications

Chapitre VI : Science naturelles et médicales

Chapitre VII : Les arts Arabes. Peinture, sculpture, arts industriels.

Chapitre VIII : L'architecture des Arabes.

Chapitre IX : Commerce des Arabes. - Leur relation avec divers pays.

Chapitre X : Civilisation de l'Europe par les Arabes. Leur influence en Occident et en Orient

Livre sixième : La décadence de la civilisation arabe.

Chapitre I : Les successeurs des arabes. - Influence des européens en Orient.

Chapitre II : Causes de la grandeur et de la décadence des Arabes. État actuel de l'islamisme.

Dans cette analyse, 797 segments ont été obtenus et deux classes de temres.ci-après on montre un extrait de ces résultats.

Segments	Contenu
Segment : 626	Nous conclurons ce chapitre en disant que la civilisation musulmane eut dans le monde une influence immense et que cette influence n'est due qu'aux Arabes et non aux races diverses qui ont adopté leur culte. Par leur influence morale, ils ont policé les peuples barbares qui avaient détruit l'empire romain ; par leur influence intellectuelle, ils ont ouvert à l'Europe le monde des connaissances scientifiques, littéraires et philosophiques qu'elle ignorait, et ont été nos civilisateurs et nos maîtres pendant six cents ans.
Segment : 595	Influence scientifique et littéraire des Arabes. - Nous allons essayer de démontrer maintenant que l'action exercée par les Arabes sur l'Occident fut également considérable, et que c'est à eux qu'est due la civilisation de l'Europe. Leur influence ne fut pas moins grande qu'en Orient, mais elle fut différente. Dans les pays orientaux elle se fit surtout sentir sur la religion, la langue et les arts. En Occident, l'influence engendrée par la religion fut nulle ; celle exercée par les arts et la langue,faible, celle résultant de l'enseignement scientifique, littéraire et moral, immense.
Segment : 29	Lorsque les Arabes s'emparèrent de la Perse et de la Syrie, ils y trouvèrent une partie du précieux dépôt de la science grecque. Sous leur influence, les versions syriaques furent traduites en arabe ; les anciens auteurs qui n'avaient pas encore été traduits le furent bientôt,

	et les études scientifiques et littéraires reçurent une impulsion très vive.
Segment : 538	L'activité commerciale des Arabes ne fut pas moindre que celle qu'ils déployèrent dans les sciences, les arts et l'industrie. À une époque où l'extrême Orient était à peine soupçonné de l'Europe, où l'Afrique, en dehors de quelques côtes, était inconnue, les Arabes étaient en relations commerciales avec l'Inde, la Chine, l'intérieur de l'Afrique et les parties les moins explorées de l'Europe telles que la Russie, la Suède et le Danemark.
Segment : 262	La découverte faite par Casiri à la bibliothèque de l'Escurial d'un manuscrit arabe sur papier de coton remontant à l'an 1009, et antérieur à tous ceux existant dans les bibliothèques de l'Europe, prouve que les Arabes furent les premiers à remplacer le parchemin par le papier.
Segment : 407	Les musées d'Europe renferment beaucoup de poteries imitées de celles des Arabes d'Espagne.L'imitation se reconnaît aisément aux fragments d'inscriptions mélangés aux ornements. Les potiers étrangers, prenant ces inscriptions pour de simples motifs d'ornementation, les ont toujours déformées en les copiant.
Segment : 73	Pour rester en paix avec la foule, les philosophes arabes avaient fini par nettement séparer la religion de la science. Leur conclusion à cet égard a été très bien formulée par le célèbre Al-Gazzali, qui enseignait au onzième siècle, à Bagdad.

Classes	Contenu
Classe : 1	…**arabe arabes**…causes celle cent cependant certain certaine certainement certaines certains cesse chacun changer chapitre chez chine chinois chose choses chrétiens chute **civilisation**…détachées déterminent développe effet… **siècle…** empires enrichir enseignement ensemble ensuite entièrement entre envoyés **espagne** esprit **europe** …hommes idéal important importantes importants inde individus indépendants **influence** …inférieures institutions intellectuelle également égypte élevé élément époque établi étude…
Classe : 2	…voies voile voiler vois voisin voisinage voisine voisines voit volant volga volon volontiers volume volumes volupté vomique voudra voudraient voudrait voudront voulaient voulait vouloir voulons… éteinte éteintes étendaient étendait étendant... étendirent étendre étendue étendues éternelle éternels éthio éthiopie…traductions traduire traduisirent traduisit traduit traduite traduites ..

À la lumière des résultats précédentes on constate que tous les membres des règles (antécédents, conséquents), présentés dans le tableau 6.7, sont tous de la même classe et que les données de la deuxième classe ne génèrent aucune règle. Rappelons que la valeur du support égale 5% et la valeur de la confiance égale 30%.

Le tableau suivant est un extrait d'une transaction (T1) qui représente le résultat de la fragmentation verticale sur la transaction globale en utilisant les termes de la première classe. Tous les termes du tableau appartiennent à la première classe.

	influence	**europe**	**siècle**	**espagne**	**arabes**	**civilisation**	**...**
S : 626	1	1	1	0	1	1	...
S : 595	1	1	0	0	1	1	...
S : 29	1	0	0	0	1	0	...
S : 538	0	0	1	0	0	1	...
S : 262	0	0	1	0	0	1	...
S : 407	0	0	1	0	0	1	...
S : 73	0	0	1	0	1	0	...

Règle d'association	Support	Confiance
influence → arabes	6%	71%
Europe → arabes	6,5%	76%
Siècle → arabes	7%	59%
Espagne → arabes	7%	70%
arabe →arabes	7,15%	47%
civilisation → arabes	7,77%	71,26%

Tableau 6.7 : résultats d'expérimentation de la partie 4.

Les résultats du tableau 6.7 démontrent une forte relation entre les mots « influence », « Europe », « Espagne » et « civilisation » et le mot « arabes » avec respectivement les confiances : 71%, 76%, 55,55% et 71,26%. On constate d'autres relations entre les mots « siècle », « arabe » et le mot « arabes » avec respectivement les confiances : 59% et 47%.

On aurait pu mettre la valeur de confiance a 10%. Avec cette valeur d'autres règles s'ajoutent à l'ensemble de règles du tableau 6.7. Le tableau 6.8 montre les nouvelles règles obtenues. On remarque que tous les mots jouent le rôle d'antécédent et de conséquent. Par conséquent, la propriété d'apparition seulement au membre droit des règles dont elle caractérise le mot « arabes » dans les règles du tableau 6.7 a disparu.

Règle d'association	Support	Confiance
Arabes → influence	6%	12,90%
Arabes → Europe	6,5%	13,70%
Arabes → Siècle	7%	14,77%

Arabes →Espagne	7%	14,77%
Arabes →arabe	7,15%	15%
Arabes → civilisation	7,77%	16,30%

Tableau 6.8 : nouvelles règles si confiance = 10%

On remarque que le mot « arabes » est un attribut de classe parce qu'il appartient à la partie droite de toutes les règles d'association résultantes. Par conséquent les règles auxquelles elles appartiennent : « influence → arabes », « Europe → arabes », « siècle → arabes », « Espagne → Arabes », « arabe → arabes » et « civilisation → arabes » s'ajouteront à l'ensemble des règles de classification [54-56].

Ainsi, l'ensemble des règles de classification et l'ensemble des attributs de classe du tableau 6.7 sont :

- Règles d'association avec contrainte = {

influence → arabes

Europe → arabes

siècle → arabes

Espagne → arabes

arabe → arabes

civilisation → arabes}

- Attributs de classe = {arabes}.

L'ensemble des mots clés est : {influence, Europe, siècle, Espagne, arabe, civilisation} Ces mots sont étroitement corrélés avec l'attribut de classe en l'occurrence « arabes ». Cet ensemble est cohérent avec les chapitres d'expérimentation. En effet, les chapitres montrent une grande influence de la civilisation arabe en Europe pendant

plusieurs siècles ainsi que l'ouverture du monde des connaissances scientifiques, littéraires et philosophiques.

Chapitre 7

CONCLUSION

Tout d'abord, nous tenons à préciser que le présent travail de recherche est en cours de publication.

Dans le cadre de ce travail, nous nous sommes intéressés aux mots clés qui fournissent des informations importantes sur le contenu du document. Notre objectif consiste à extraire ces mots à partir d'un ou de plusieurs textes. Pour ce faire, nous avons développé un prototype logiciel qui analyse des textes en combinant à la fois la classification et l'extraction des règles d'association régulières.

Avant d'aborder ce problème, nous avons présenté les différentes méthodes de classification les plus connues qui ont été citées dans la littérature scientifique. Ensuite, nous avons présenté les concepts mathématiques du modèle vectoriel et le schéma de pondération TF-IDF.

Après avoir rappelé les règles d'association, leurs concepts de base et l'algorithme Apriori, nous avons proposé notre méthode de résolution du problème. Cette méthode est une chaine de traitement qui se déroule en trois phases et qui se base principalement sur la classification des vecteurs de termes et l'application de l'algorithme Apriori sur chacune des classes résultantes.

Finalement, nous avons décrit l'architecture du système et ses principales interfaces utilisateur. Nous avons, également, procédé à l'expérimentation en analysant un livre complet à savoir « La civilisation des Arabes de Gustave Le Bon (1884) ».

À travers les expérimentations qui ont été effectués dans le cadre de ce mémoire, nous avons mis en lumière des résultats significatifs :

i. L'extraction des règles d'association régulières à partir des résultats de la classification nous a permis de révéler l'importance de l'identification des attributs de classe dans le processus de sélection des mots clés. Ces derniers sont fortement corrélés avec ce type d'attribut. Nous avons démontré, également, que la recherche des attributs de classe dépend fortement au choix des seuils (support, confiance). Par conséquent, le bon choix de ces paramètres conduit à un résultat de qualité.

ii. La plupart des règles obtenues sont des règles de classification qu'on peut utiliser pour élaborer un modèle de classification basée sur ces règles.

Malgré les bons résultats obtenus, on remarque que le choix des seuils (support, confiance) reste une étape critique et décisive, pour cela, et dans une perspective d'amélioration de notre approche, nous nous proposons :

- De prendre en compte la particularité de chaque classe dans le choix des seuils.
- Développer une heuristique pour un choix dynamique des seuils.
- Généraliser notre prototype logiciel dans le but de tester notre approche sur différentes langues.

Bibliographie

3. Hilali, h., *application de la classification textuelle pour l'extraction des règles d'association maximales.* thèse de maitrise en informatique, université du québec à trois-rivières, trois-rivières, 2009.
4. Descôteaux, s., *les règles d'association maximale au service de l'interprétation des résultats de la classification.* thèse de maitrise en informatique, université du québec à trois-rivières, trois-rivières, 2014: p. 174.
5. Du, t. and j. dy, *a deterministic method for initializing k-means clustering.*
12. Gurney, k., *an introduction to neural networks.* 1997.
13. Fausett, l.v., *fundamentals of neural networks: architectures, algorithms and applications.* 1993.
17. Veloso, m., *note du cours , < perceptrons and neural networks >,2001.*
19. Goodfellow, i., y. bengio, and a. courville, *deep learning.* mit press, 2016.
22. Amira, l., b. khaoukha, and l. adila, *identification et commande des systèmes non linéaires.* 2011.
24. Guthikonda, s.m., *kohonen self-organizing maps* december 2005.
26. Ritter, h., t. martinetz, and k. schulten, *neural computation and self-organizing maps; an introduction.* 1992.
27. Laouamer, l., *approche exploratoire sur la classification appliquée aux images.* 2006.
29. Bokhabrine, y., *etude et comparaison d'algorithmes d'optimisation pour la reconstruction 3d par supershapes et r-fonctions. .* 2006.
30. Vallée, t. and m. yıldızoglu, *présentation des algorithmes génétiques et de leurs applications en économie.* 2001.
31. Michalewicz, z., *genetic algorithm + data structures = evolution programs.* 1994.
32. Deng, l. and d. yu, *deep learning methods and applications.* 2014.
36. Salton, g. and c. buckley, *term-weighting approaches in automatic text retrieval.* inf. process. manage., 1988. **24**(5): p. 513-523.
37. Salton, g., a. wong, and c.s. yang, *a vector space model for automatic indexing.* commun. acm, 1975. **18**(11): p. 613-620.
38. Manning, c.d., p. raghavan, and h. schutze, *introduction to information retrieval.* irbook, may 27, 2008.
41. Monbet, v., *chapitre 5. règles d'association.* université de rennes 1, 15-nov-2010
42. Agrawal, r. and r. srikant, *fast algorithms for mining asso ciation rules.* proceedings of the 20th vldb conference santiago, 1994.
43. Agrawal, r., t. imielinski, and a. swami, *mining association rules between sets of items in large databases.* acm sigmod conference washington, 1993.
44. Han, j. and m. kamber, *data mining: concepts and techniques.* 2006: p. 761.
45. Sharma, n. and d.c.k. verma, *association rule mining: an overview* ijcsc volume 5 number 1 2014.

46. Moreno, m.n., s. segrera, and v.f. lópez, *association rules: problems, solutions and new applications* actas del iii taller nacional de minería de datos y aprendizaje, 2005.
47. García, e., et al., *drawbacks and solutions of applying association rule mining in learning management systems*
48. Nourashrafeddin, s.n., e. milios, and d. arnold, *interactive text document clustering using feature labeling.* acm, 2013.
54. Sun, y. and a.k.c. wong, *an overview of associative classifiers.* ieee
55. Wang, g. and q. song, *a novel feature subset selection algorithm based on association rule mining.* ieee, 2013.
56. Xie, j., j. wu, and q. qian, *feature selection algorithm based on association rules mining method.* ieee, 2009.

Webographie

1. Wikipedia, *k-means clustering,* <https://en.wikipedia.org/wiki/k-means_clustering >,*2017.*
2. Xlstat, *classification par la méthode de k-means,* <https://www.xlstat.com/fr/solutions/fonctionnalites/classification-par-la-methode-des-nuees-dynamiques-k-means >.
6. Thirumuruganathan, s., *a detailed introduction to k-nearest neighbor (knn) algorithm* , <https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/>, *may 2010.*
7. Wikipedia, *k-nearest neighbors algorithm,* < https://en.wikipedia.org/wiki/k-nearest_neighbors_algorithm >,*2016.*
8. Sabrina, t., *apprentissage automatique et réduction du nombre de dimensions* ,<http://www-ia.lip6.fr/~tollaris/articles/these/node7.html>,2008.
9. Tay, b., j.k. hyun, and s.o.s. oh, *computational and mathematical methods in medicine* ,<https://www.researchgate.net/figure/260397165_fig7_pseudocode-for-knn-classification>, *jan 2014.*
10. Wikipedia, *machine à vecteurs de support* ,<https://fr.wikipedia.org/wiki/machine_%c3%a0_vecteurs_de_support >, *2017.*
11. Learn, s., *support vector machines* ,<http://scikit-learn.org/stable/modules/svm.html >.
14. Däullary, t., *deep stacking networks* , <http://recognize-speech.com/acoustic-model/knn/comparing-different-architectures/deep-stacking-networks>, *21 december 2014.*
15. Krüger, s., *introduction to artificialneural networks* < http://recognize-speech.com/basics/introduction-to-artificial-neural-networks >,*decembre 2014.*
16. Wikipedia, *perceptron* , < https://fr.wikipedia.org/wiki/perceptron >.
18. Wikipedia, *réseau de neurones de hopfield* , <https://fr.wikipedia.org/wiki/r%c3%a9seau_de_neurones_de_hopfield>.
20. Wikipedia, *perceptron multicouche,* < https://fr.wikipedia.org/wiki/perceptron_multicouche >.
21. Wikiversite, *réseaux de neurones : avantages et possibilités* , <https://fr.wikiversity.org/wiki/r%c3%a9seaux_de_neurones/avantages_et_possibilit%c3%a9s>,18 *mai 2016.*
23. Wikipedia, *self-organizing map* , < https://en.wikipedia.org/wiki/self-organizing_map >,*2017.*
25. Stanford, *neural networks* ,<http://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/architecture/variations.html >.
28. Wikipedia, *algorithme génétique* , <https://fr.wikipedia.org/wiki/algorithme_g%c3%a9n%c3%a9tique>.

33. Wikipedia, *apprentissage profond* , <https://fr.wikipedia.org/wiki/apprentissage_profond>.
34. Press, c.u., *evaluation of clustering,* <http://nlp.stanford.edu/ir-book/html/htmledition/evaluation-of-clustering-1.html >,*2008*
35. Stanford, *autoencoders,* <http://ufldl.stanford.edu/tutorial/unsupervised/autoencoders/>.
39. Wikipedia, *tf-idf* , < https://fr.wikipedia.org/wiki/tf-idf >.
40. Wikipedia, *vector space model* , < https://en.wikipedia.org/wiki/vector_space_model >.
49. Uqac, *la civilisation des arabes* ,<http://classiques.uqac.ca/classiques/le_bon_gustave/civilisation_des_arabes/civilisation_arabes.html > , *2008.*
50. Wikipedia, *f sharp* , < https://en.wikipedia.org/wiki/f_sharp_(programming_language*)* >.
51. Wikipedia, *strong and weak typing* , < https://en.wikipedia.org/wiki/strong_and_weak_typing#definitions_of_.22strong.22_or_.22weak.22 >.
52. Wikipedia, *programmation fonctionnelle* , <https://fr.wikipedia.org/wiki/programmation_fonctionnelle >.
53. Wikipedia, *programmation impérative* , < https://fr.wikipedia.org/wiki/programmation_imp%c3%a9rative >.