

CHAPITRE 5

REALISATION ET EXPERIMENTATION

5.1. Introduction :

Ce chapitre est essentiellement consacré aux grandes lignes qui visent à réaliser l'objectif de ce thème, et les outils exploités pour le développement du logiciel tels que le choix du langage de programmation, l'environnement de programmation, le matériel utilisé, les principales fonctions et la structure générale du logiciel.

5.2. Présentation des corpus utilisés :

Notre base de test et d'apprentissage basée sur des codes de catégories choisis avec un nombre de documents (en anglais, en arabe et en français) par catégorie représenté comme suit :

		Les langues					
		Anglais		Arabe		Français	
Code	Catégorie	apprentissage	teste	apprentissage	teste	apprentissage	teste
1	Art	05	03	03	02	07	05
2	Economie	04	03	04	02	04	03
3	Politique	03	03	04	01	03	03
4	Sport	02	02	03	02	04	05

Tableau 5.1 : Définition des catégories, avec le nombre de documents disponibles pour chaque partie de la base.

- A partir ce tableau Nous avons 80 fichiers textes.
- Les fichiers sont enregistrés dans l'encodage UTF-8.

5.3. Représentation des textes et l'approches utilisées:

Toutes nos approches utilisent une représentation « sacs de mots » comme méthode de représentation, issue du modèle vectoriel. Etant donné le grand nombre de descripteurs potentiels, il est, en général, nécessaire d'effectuer une sélection de descripteurs avant de pouvoir utiliser un modèle d'apprentissage.

Cette technique originale de sélection de descripteurs en deux étapes présente plusieurs avantages. Elle est entièrement automatique et ne nécessite pas de ressources externes (comme une

liste de mots les plus fréquents dans une langue donnée) et elle est couplée avec un critère d'arrêt pour trouver le "bon" nombre de descripteurs.

Nous utilisons comme méthodes d'apprentissage :

L'algorithme arbre de décision, la méthode la plus simple et la plus populaire pour la catégorisation thématique des textes. On a choisi pour la construction des nos arbres C4.5, l'algorithme la plus utilisées dans la communauté de l'apprentissage automatique et pour l'identification de la langue l'algorithme Naïve de Bayes puisqu'il donne de très bons résultats.

Les documents seront représentés par des vecteurs de mots sans utilisation de ressource sémantique.

Une fois les documents représentés, ils seront pondérés afin de donner un poids pour chaque mot dans chaque document. La pondération utilisée est la pondération TF.

Par la suite, nous utilisons nos approches pour classer les nouveaux documents qui doivent être représentés et pondérés.

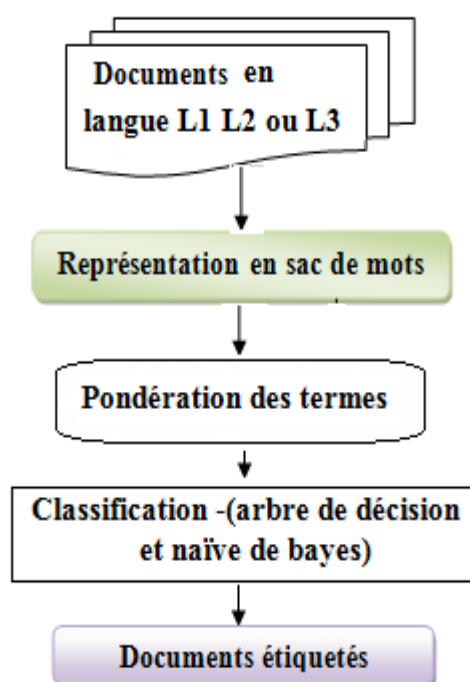


Figure 5.1 : Processus de l'arbre de décision.

L1 : Français.

L2 : Anglais.

L3 : Arabe.

5.4. Application de l'algorithme d'arbre de décision (C4.5) dans la catégorisation des documents : [40]

Pour classer des documents dans des catégories, il faut construire un arbre de décision par catégorie. Et pour déterminer à quelle catégorie appartient un nouveau document, on utilise l'arbre de décision de chaque catégorie pour classer ce document.

Chaque arbre répond par Oui si le document appartient à la catégorie et par Non dans le cas échéant (il prend une décision).

Concrètement, chaque nœud d'un arbre de décision contient un test (un IF...THEN) et les feuilles ont les valeurs Oui ou Non. Chaque test regarde la valeur d'un attribut de chaque exemple. En effet, on suppose qu'un exemple est un ensemble d'attributs/valeurs. Pour des documents, chaque attribut peut être un mot, et la valeur sera par exemple 0 ou 1 selon que ce mot appartient ou non au document.

Pour construire l'arbre de décision, il faut trouver quel attribut on doit prendre à chaque nœud. C'est un processus récursif. Et pour cela, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples Oui/Non.

On crée alors un nœud contenant ce test, et on crée autant de descendants que de valeurs possibles pour ce test.

Exemple : si on teste la présence d'un mot, les valeurs possibles sont Présent/Absent. A chaque fois, on aura donc deux descendants pour chaque nœud.

On répète ce processus en associant à chaque descendant le reste des exemples qui satisfont le test du prédécesseur.

Si toutes ou la grande majorité des instances de T_f ont la même classe $c \in \{1, \dots, C\}$, c apparaît comme la meilleure prédiction possible, et la fraction des instances de classe c dans T_f est un indicateur de la « sûreté » de cette prédiction. Dans le cas contraire, prédire la classe la plus fréquente reste une option, mais la sûreté de la prédiction n'en sera que moins bonne.

5.5. Application de l'algorithme de bayes naïve sur la détection de la langue :

L'idée est d'utiliser des conditions de probabilité observées dans les données. On calcule la probabilité de chaque classe parmi les exemples. Ce sont les "prior probabilities". Par exemple, si la classe "politique" revient 2 fois sur les 5 documents donnés en exemple, sa "prior probability" sera de $2/5$. En plus des "prior probas", l'algorithme calcule les fréquences d'apparition de chaque variable d'entrée avec celles de sortie. Pour classer des documents, les variables d'entrée sont les mots présents dans l'ensemble des documents. A chaque mot on calcule le nombre de fois qu'il

apparaît dans les documents classés dans une classe donnée. On calcule cette fréquence pour chaque classe et la classe ayant la plus grande valeur de la fréquence est la classe qui notre document appartienne.

5.6. Environnement matériel et logiciel :

5.6.1. Configuration matérielle et logicielle :

Nos expérimentations ont été développées sur une machine ayant les caractéristiques suivantes :

- Un PC Pentium 4 à un processeur Intel (R) Celeron(R) CPU B820@ 1.70Ghz 1.70Ghz.
- Une mémoire de 2GO.
- L'ensemble est piloté par le système d'exploitation Windows7.

Les outils et langages utilisés pour la manipulation des données sont : Microsoft Office 2007 Professionnel et JAVA (NetBeans IDE 6.8).

5.6.2. Le langage de programmation (JAVA) :

Notre choix pour le langage de programmation s'est porté sur le langage JAVA, et cela parce qu'il est un langage orienté objet simple ce qui réduit les risques d'incohérence et il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers ainsi que beaucoup de fonctionnalités qui peuvent être utilisées pour développer des applications diverses. Il existe une multitude de bibliothèques développées et fournies pour être utilisées en JAVA. Les API (Application Programming Interface) des autres langages autres que JAVA ne sont pas finalisées et doivent encore être mises à jour [03].

5.6.3. L'environnement de programmation :

NetBeans IDE est un environnement de développement intégré (EDI) de premier ordre pour Windows, Mac, Linux et Solaris. Le projet NetBeans consiste en un EDI Open Source et en une plate-forme d'application permettant aux développeurs de créer rapidement des applications Web, d'entreprise, de bureau et mobiles à l'aide de la plate-forme Java, ainsi que de Java FX, PHP, JavaScript et Ajax, Ruby et Ruby on Rails, Groovy et Grails et C/C++.

Le projet NetBeans est soutenu par une communauté de développeurs dynamique et propose différentes ressources de documentation et de formation, ainsi qu'un choix varié de plug-ins tiers.

NetBeans IDE 6.8 est le premier EDI à prendre en charge l'intégralité de la spécification Java EE 6, avec prise en charge améliorée de JSF 2.0/Facelets, Java Persistence 2.0, EJB 3.1 et notamment l'utilisation des EJB dans des applications Web, les services Web RESTful et GlassFish v3. Nous

recommandons également cette version pour développer avec la dernière version de JavaFX SDK 1.2.1 et pour créer des applications Web PHP avec la nouvelle version de PHP 5.3 ou avec le framework Symfony.

Notre intégration unique de Project Kenai, un environnement collaboratif permettant d'héberger des projets Open Source, offre désormais la prise en charge complète de JIRA, ainsi que l'intégration d'un programme de messagerie instantané amélioré et d'un outil de suivi des problèmes. Nous avons également ajouté des fonctionnalités à l'intégration des bases de données et Maven de l'EDI et avons amélioré l'intégration de l'éditeur et des outils des projets Ruby, Groovy et C/C++.

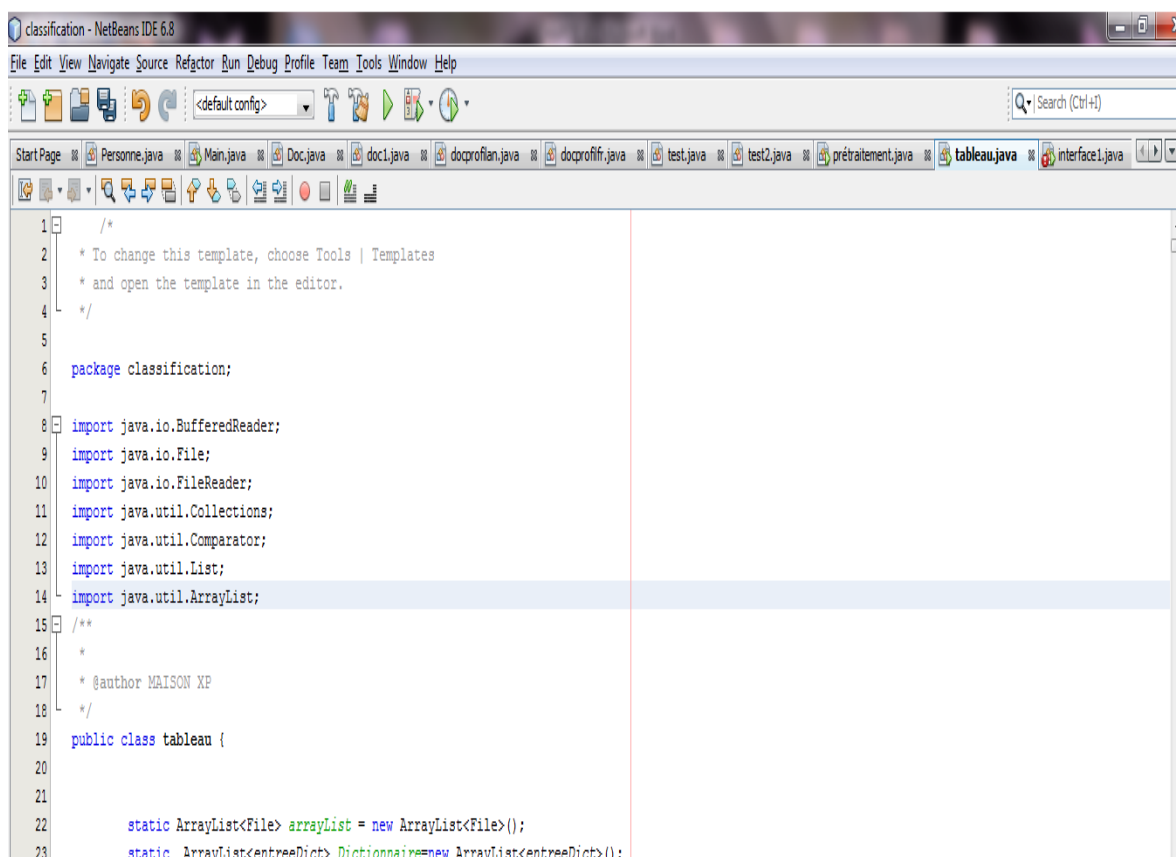


Figure 5.2 : L'interface graphique de NetBeans IDE 6.8.

5.7. Structure et fonctionnement de l'application :

5.7.1. Structure du classifieur :

Notre prototype se compose d'une fenêtre principale à partir de laquelle l'utilisateur peut effectuer les opérations ou les traitements désirés en sélectionnant un élément du menu ou en cliquant sur un bouton.

Cette fenêtre est montrée dans la figure suivante :

Interface du logiciel:

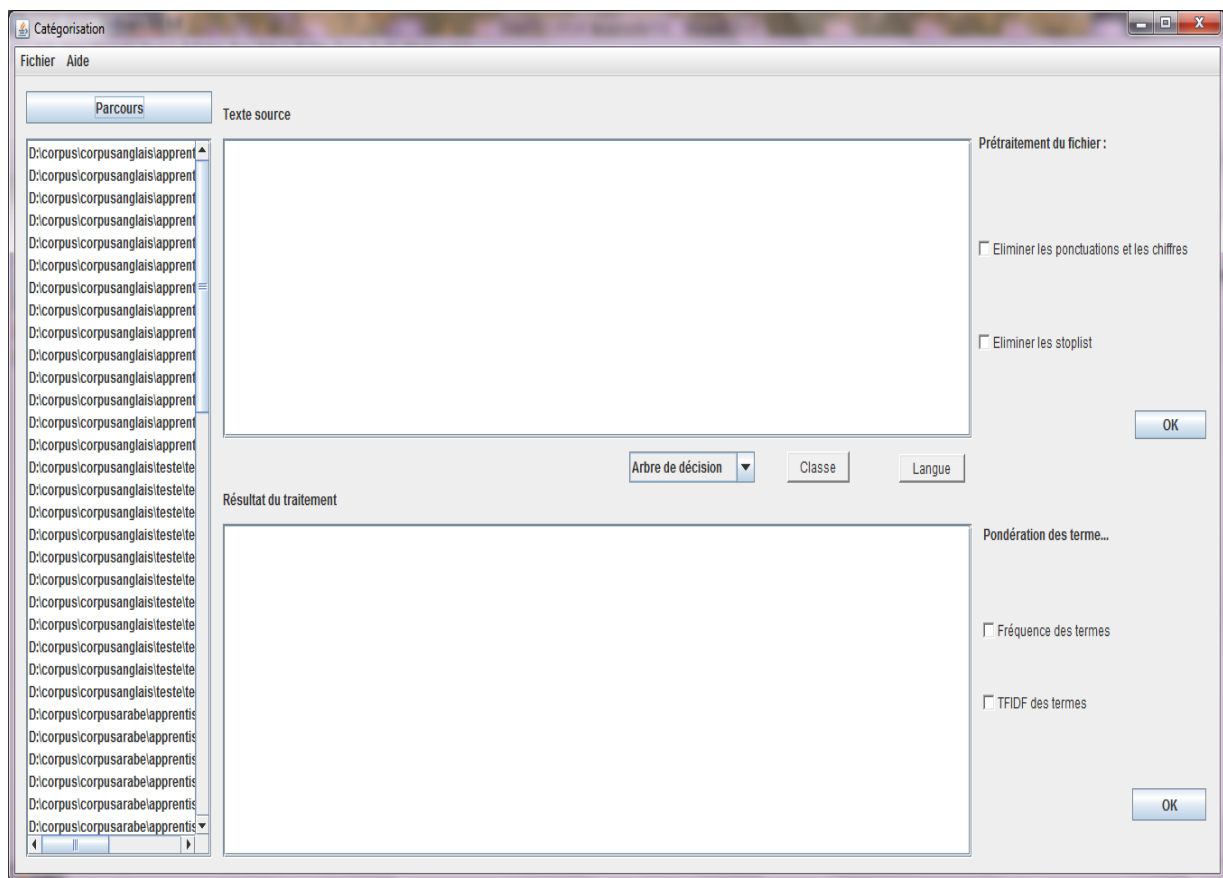


Figure 5.3 : Interface du logiciel.

Comme le montre la figure, notre application comporte :

- Un menu principal : Fichier, Aide.
- Deux boutons : Classe et Langue.
- Des boites à options (CheckBox) : pour le choix prétraitements ou pondération des termes.
- Des zones blanche : (Texte source et Résultat des traitements) pour l'affichage des résultats.

Effectuer des prétraitements sur un fichier:

- Choisir un fichier :

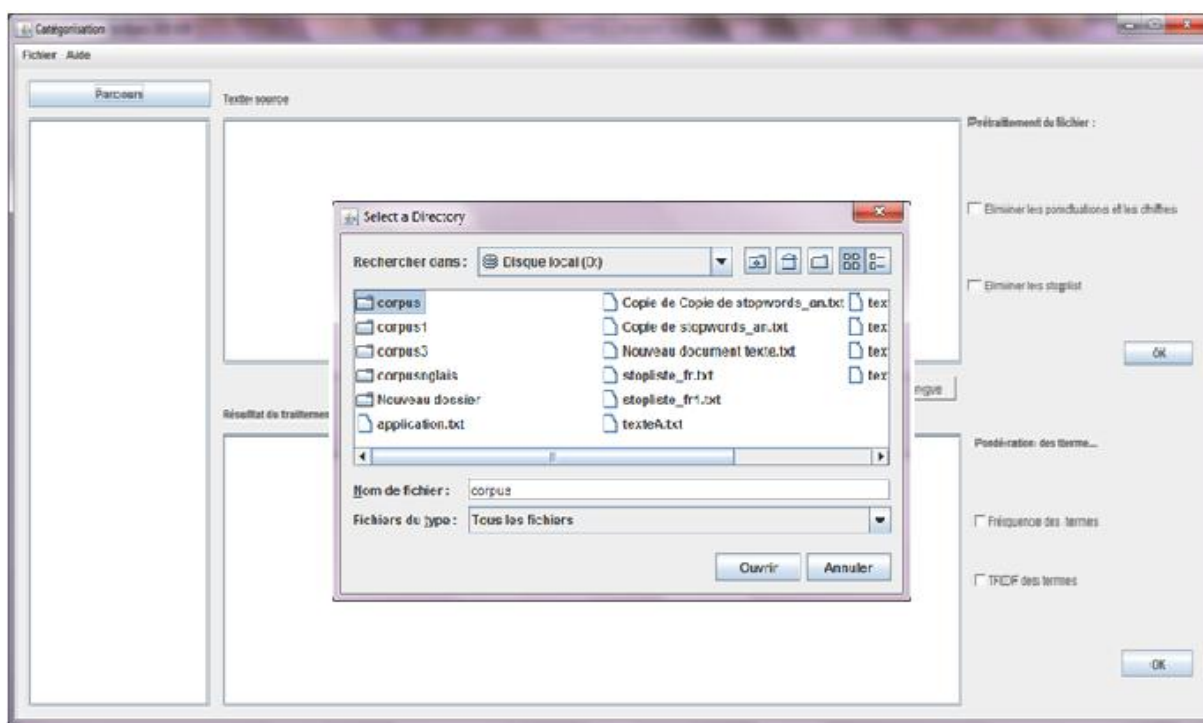


Figure 5.4 : ouvrir un dossier des fichiers.

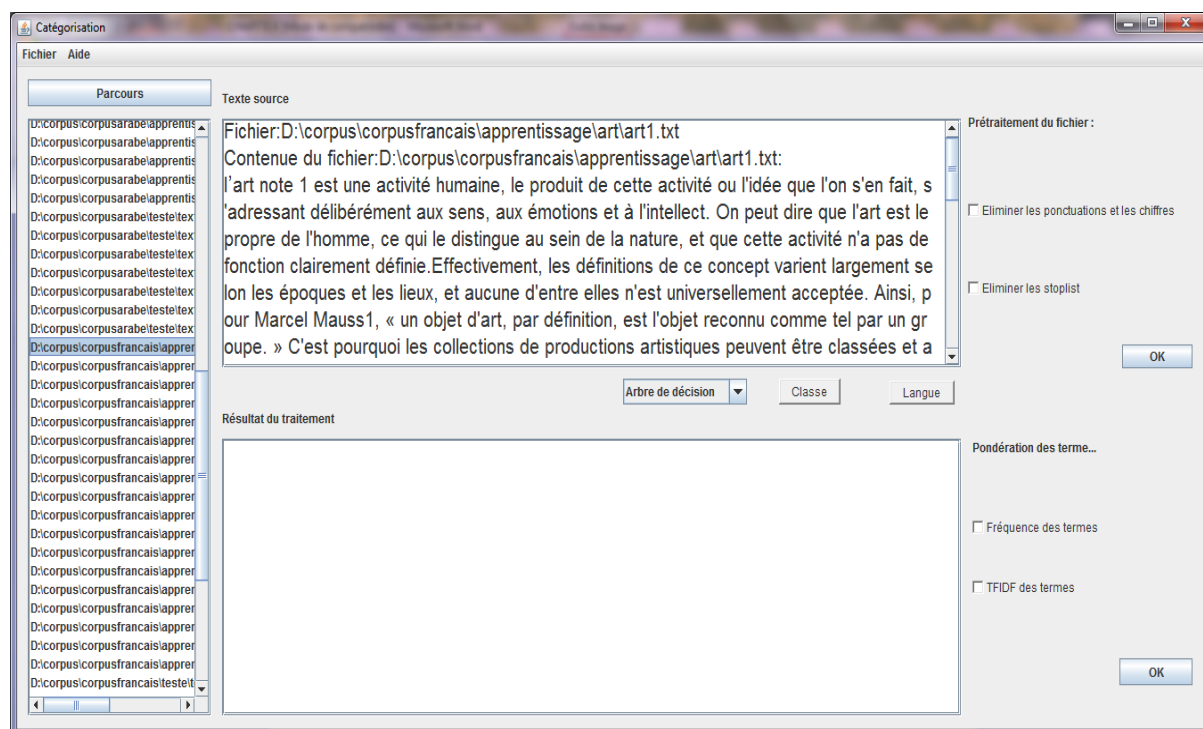


Figure 5.5 : Sélectionner un fichier.

Pondération des termes du fichier :

- Fréquence des termes du fichier :

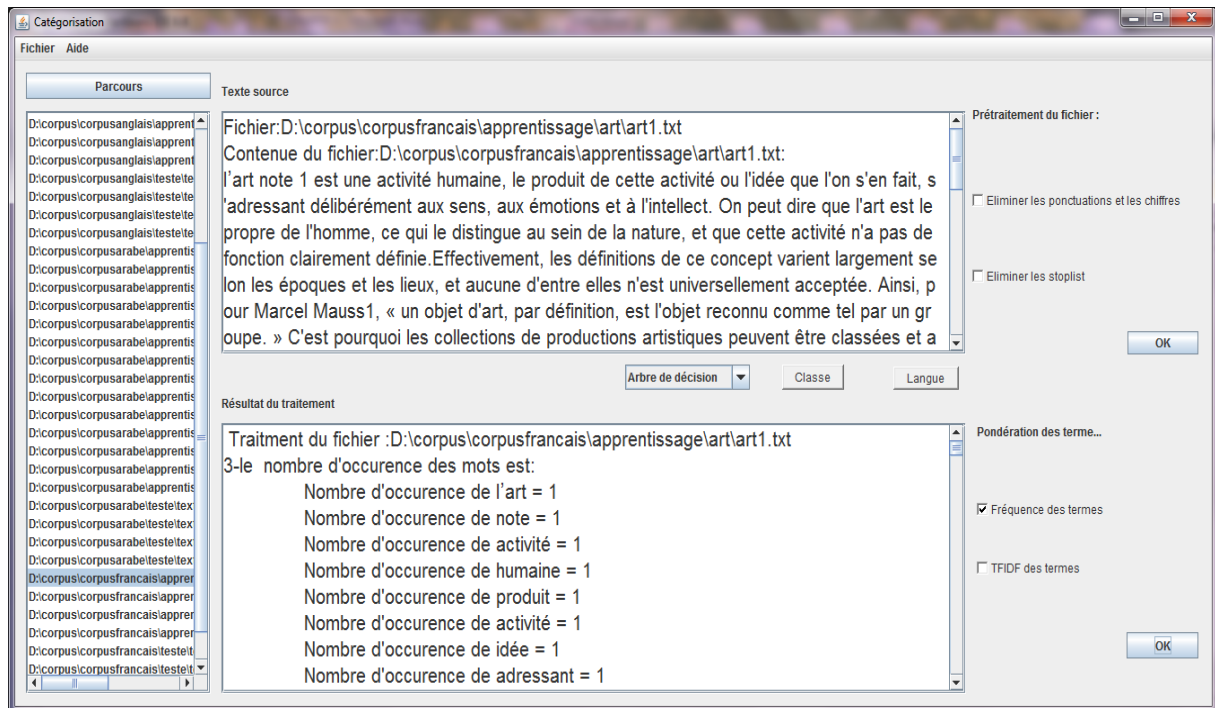


Figure 5.8 : Fréquence des termes.

- TFIDF des termes du fichier :

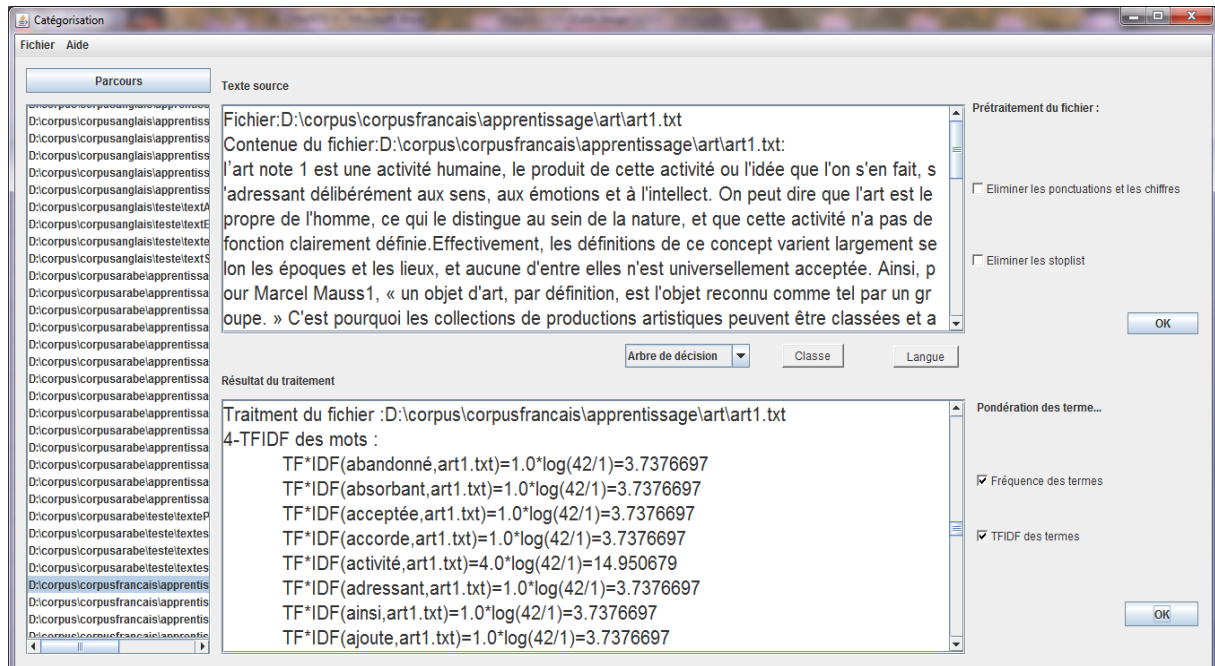


Figure 5.9 : TFIDF des termes.

Un test avec la méthode de naïve bayes :

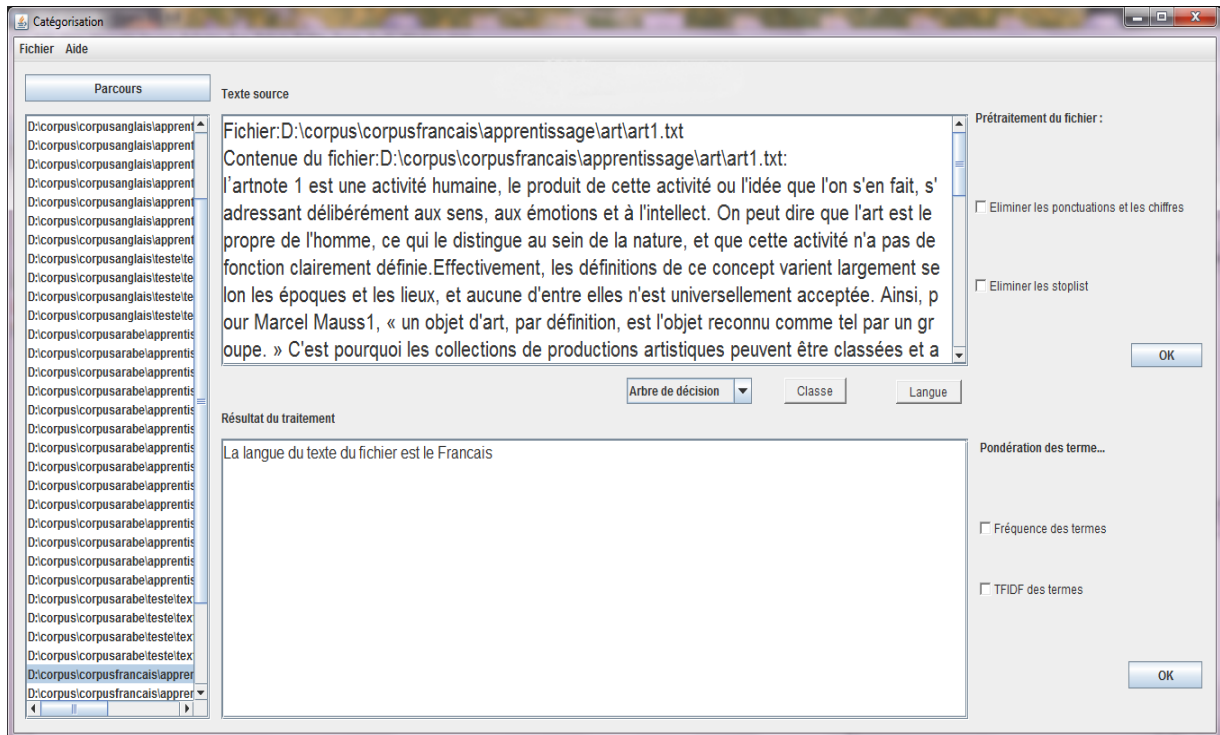


Figure 5.10 : Identification de la langue du texte.

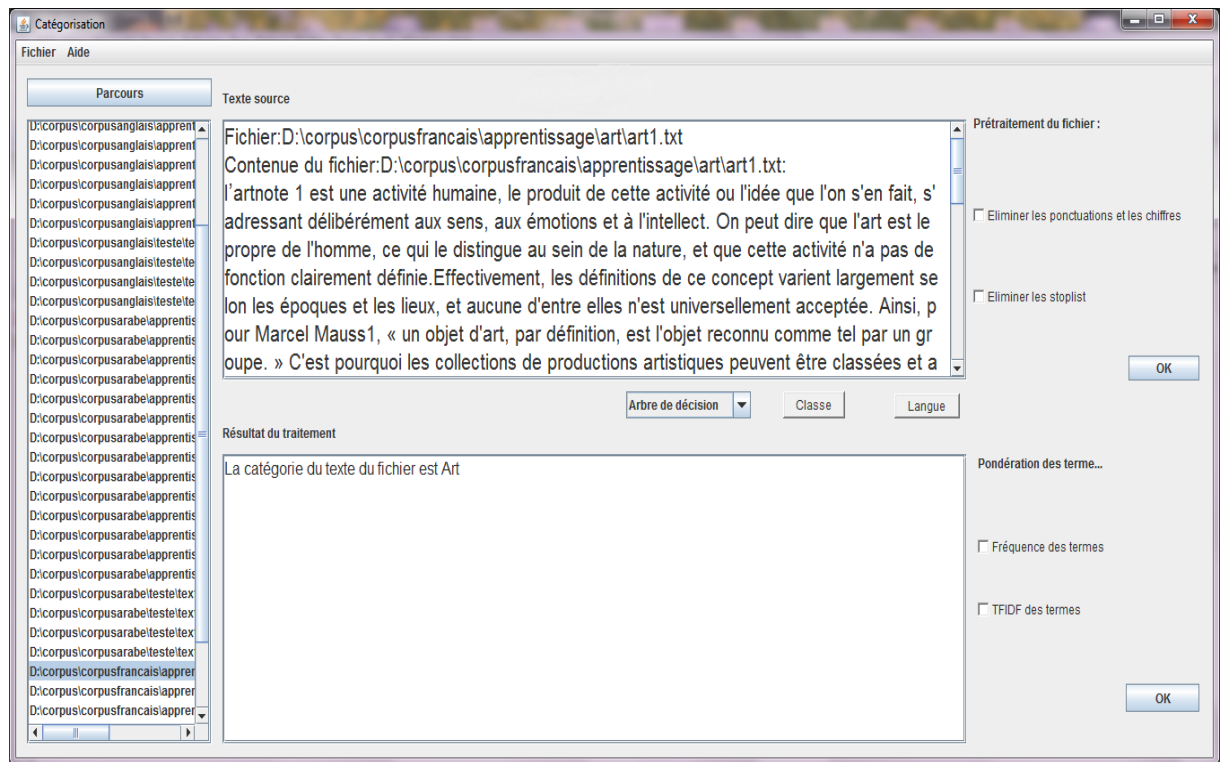


Figure 5.11 : Catégorisation du texte.

5.7.2. Fonctionnement du logiciel :

L'architecture suivante représente le fonctionnement de notre application. Son fonctionnement est composé de plusieurs étapes qui peuvent être visualisées une par une dans la figure suivante :

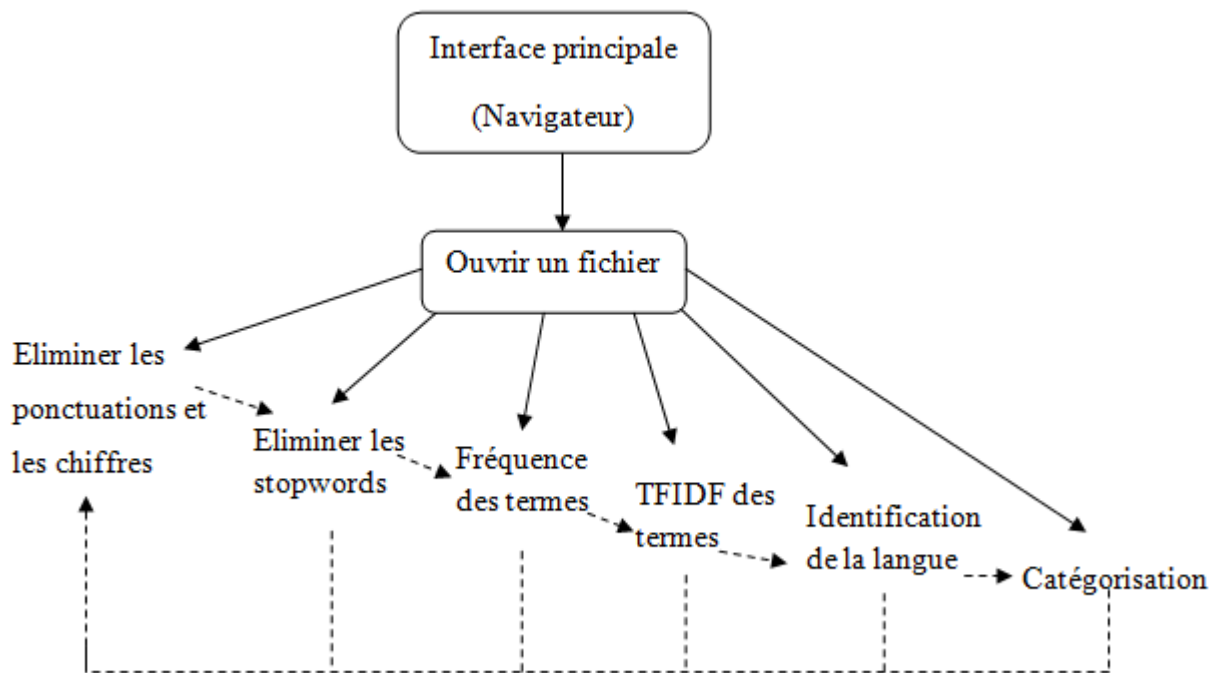


Figure 5.12 : Fonctionnement du logiciel.

—————> : Effectuer une opération.

-----> : Opération par défaut.

On peut résumer le fonctionnement par les étapes suivantes :

- **Etape1 :** L'utilisateur ouvre l'interface principale (Navigateur, Explorateur).
- **Etape2 :** L'utilisateur ouvre un fichier local sur le disque dur à partir du menu fichier par l'élément « ouvrir » ou à partir le bouton « parcours » où il peut ouvrir un dossier des fichiers puis choisir un fichier avec l'option sélectionne puis lui affiche.
- **Etape3 :** L'utilisateur peut faire plusieurs opérations sur le fichier à choisi :
 - ✓ Des prétraitements : élimination des ponctuations, chiffres, caractères spéciaux et stoplist via le coché sur un des cases à cochés (Eliminer les ponctuations et les chiffres ou Eliminer les stoplist).
 - ✓ Pondération des termes (soit avec la fréquence de chaque mot via l'élément Fréquence des termes ou avec le tfidf des termes via la case à coché TFIDF des termes).

- ✓ Identification de la langue de ce fichier à partir le bouton Langue (fichier de langue anglais, français ou arabe).
- ✓ La catégorie de ce fichier après choisir un algorithme de la liste des algorithmes de catégorisation proposé dans notre cas il ya deux algorithmes qui fonctionne arbres de décision et naïve de bayes (fichier de catégorie art, politique, économie ou sport).
- **Etape4** : Lorsque l'utilisateur choisi l'arbre de décision ou naïve de bayes comme méthode d'apprentissage, l'algorithme transforme le vecteur d'occurrences de texte à classer en vecteur de fréquences selon le codage de la méthode.
- **Etape5** : L'algorithme calcule la similarité entre le texte à classer et les profils prototypiques des quatre classes (art/politique/économique/sport) de l'algorithme à choisir.
- **Etape6** : Selon la métrique de prise de décision de l'algorithme choisi, le fichier sera classé.

5.8. Evaluation des résultats du classifieur :

5.8.1. Evaluation de l'algorithme Bayes de naïve (détection de langue):

On a utilisé un corpus constitué de 18 textes d'apprentissage et 16 textes de test pour la langue français, 14 textes d'apprentissage et 11 textes de test pour l'anglais et 14 textes d'apprentissage plus 7 textes de test pour l'arabe, dont on connaît à priori leurs langue, comme c'est expliqué dans les tableaux suivants :

Exemple d'expérimentation1 : (français)

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	15	01
Documents rejets à la catégorie par le classifieur	02	17

Tableau 5.2 : Table de contingences pour l'algorithme *Bayes de naïve* (français).

- **Résultat d'évaluation** :
 - **Précision** : $P = 0.9375$.
 - **Rappel** : $R = 0.8823$.
 - **F-mesure** = 0.90.

Exemple d'expérimentation2 : (anglais)

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	09	02
Documents rejets à la catégorie par le classifieur	09	00

Tableau 5.3 : Table de contingences pour l'algorithme *Bayes de naïve (anglais)*.

- **Résultat d'évaluation :**

- **Précision : P = 0.82.**
- **Rappel : R = 0.5.**
- **F-mesure = 0.62.**

Exemple d'expérimentation3 : (arabe)

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	07	00
Documents rejets à la catégorie par le classifieur	01	08

Tableau 5.4 : Table de contingences pour l'algorithme *Bayes de naïve (arabe)*.

- **Résultat d'évaluation :**

- **Précision : P = 1.**
- **Rappel : R = 0.875.**
- **F-mesure = 0.93.**

- **Taux_Correct =0.911**
- **Taux_Erreur =0.09**

❖ **Interprétation des résultats obtenus :**

Les valeurs de : Précision, Rappel, F-mesure fournies par l'algorithme naïve de bayes pour les classes «Français», «Anglais» et «Arabe» sont égaux respectivement à (0.9375, 0.8823, 0.90) pour le français, (0.82, 0.5, 0.62) pour l'anglais et (1, 0.875, 0.93) pour l'arabe. Ces mesures donnent des valeurs très élevées car ils sont plus proche à la valeur 1(proche à 100% de succès), cela veut dire que la catégorisation faite par l'algorithme Naïve Bayes pour l'identification de la langue fournie des meilleurs résultats grâce à leur calcul probabiliste que se base, c-à-dire on constate toujours que naïve bayes donne les meilleurs valeurs pour tous les paramètres d'évaluation.

5.8.2. Evaluation de l'algorithme Arbre de décision (catégorisation) :

On a utilisé un corpus constitué de 18 textes d'apprentissage et 16 textes de test pour la langue français, 14 textes d'apprentissage et 11 textes de test pour l'anglais et 14 textes d'apprentissage plus 7 textes de test pour l'arabe, dont on connaît à priori leurs langues, comme c'est expliqué dans les tableaux suivants :

▪ **Langue: Français****Exemple d'expérimentation1 : (Art)**

5 fichiers de type art.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	05	00
Documents rejets à la catégorie par le classifieur	00	05

Tableau 5.5 : Table de contingences pour l'algorithme *arbre de décision* (art).

• **Résultat d'évaluation :**

- **Précision : P =1**
- **Rappel : R =1**
- **F-mesure = 1**

➤ **Exemple d'expérimentation2 : (économie)**

3 fichiers de type économie.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	02	01
Documents rejets à la catégorie par le classifieur	00	02

Tableau 5.6 : Table de contingences pour l'algorithme *arbre de décision (économie)*.

- **Résultat d'évaluation :**

- Précision : $P = 0.67$.
- Rappel : $R = 1$.
- F-mesure = 0.80 .

➤ **Exemple d'expérimentation3 : (politique)**

3 fichiers de type politique.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	02	01
Documents rejets à la catégorie par le classifieur	01	03

Tableau 5.7 : Table de contingences pour l'algorithme *arbre de décision (politique)*.

- **Résultat d'évaluation :**

- Précision : $P = 0.67$.
- Rappel : $R = 0.67$.
- F-mesure = 0.67 .

➤ **Exemple d'expérimentation4 : (sport)**

5 fichiers de type sport.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	05	00
Documents rejets à la catégorie par le classifieur	01	06

Tableau 5.8: Table de contingences pour l'algorithme *arbre de décision (sport)*.

• **Résultat d'évaluation :**

- Précision : $P = 1$
- Rappel : $R = 0.84$.
- F-mesure = 0.90 .

Taux_Correct = 0.875

Taux_Erreur = 0.125

Exemples des modèles d'arbres :

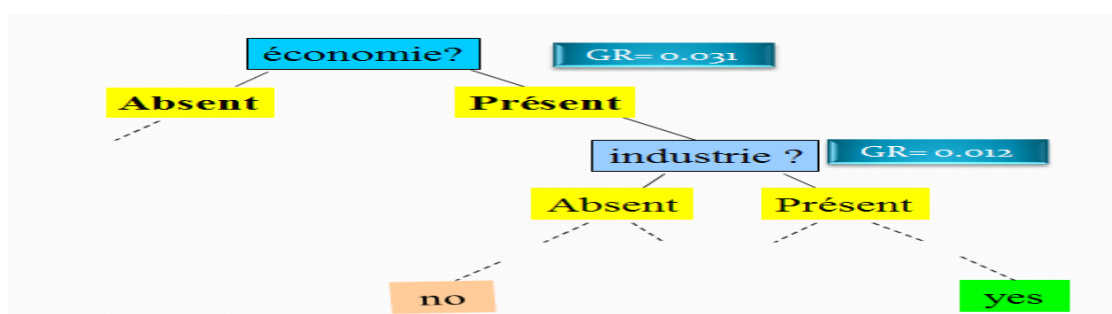


Figure 5.12 : un arbre de décision pour la catégorie économie (français)

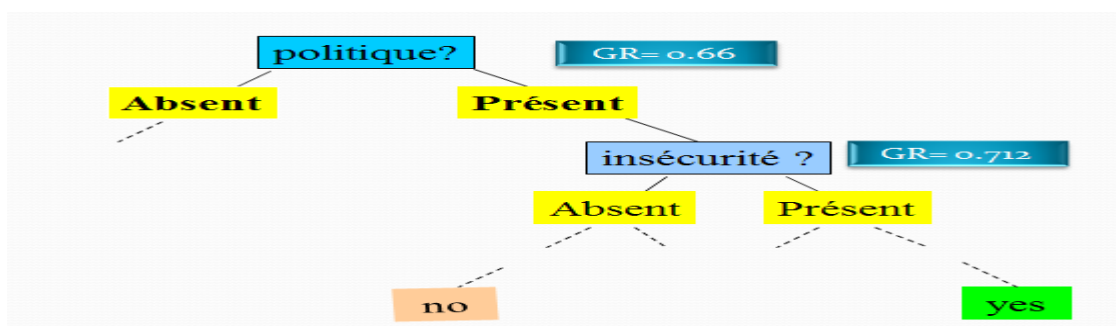


Figure 5.13 : un arbre de décision pour la catégorie politique (français)

▪ **Langue: Anglais.**

Exemple d'expérimentation1 : (Art)

3 fichiers de type art.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	02	01
Documents rejets à la catégorie par le classifieur	00	02

Tableau 5.9: Table de contingences pour l'algorithme *arbre de décision (Art)*.

• **Résultat d'évaluation :**

- Précision : $P = 0,67$
- Rappel : $R = 1$
- F-mesure = $0,80$

Exemple d'expérimentation2 : (économie)

3 fichiers de type économie.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	03	00
Documents rejets à la catégorie par le classifieur	02	05

Tableau 5.10: Table de contingences pour l'algorithme *arbre de décision (économie)*.

• **Résultat d'évaluation :**

- Précision : $P = 1$
- Rappel : $R = 0,60$
- F-mesure = $0,75$

Exemple d'expérimentation3 : (politique)

3 fichiers de type politique.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	03	00
Documents rejets à la catégorie par le classifieur	00	03

Tableau 5.11: Table de contingences pour l'algorithme *arbre de décision* (politique).

- **Résultat d'évaluation :**
 - **Précision : $P = 1$**
 - **Rappel : $R = 1$**
 - **F-mesure = 1**

Exemple d'expérimentation4 : (sport)

2 fichiers de type sport.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	01	01
Documents rejets à la catégorie par le classifieur	00	01

Tableau 5.12: Table de contingences pour l'algorithme *arbre de décision* (sport).

- **Résultat d'évaluation :**
 - **Précision : $P = 0,5$**
 - **Rappel : $R = 1$**
 - **F-mesure = 0,67**
- Taux_Correct = 0,81**
- Taux_Erreur = 0.182**

Exemples des modèles d'arbres :

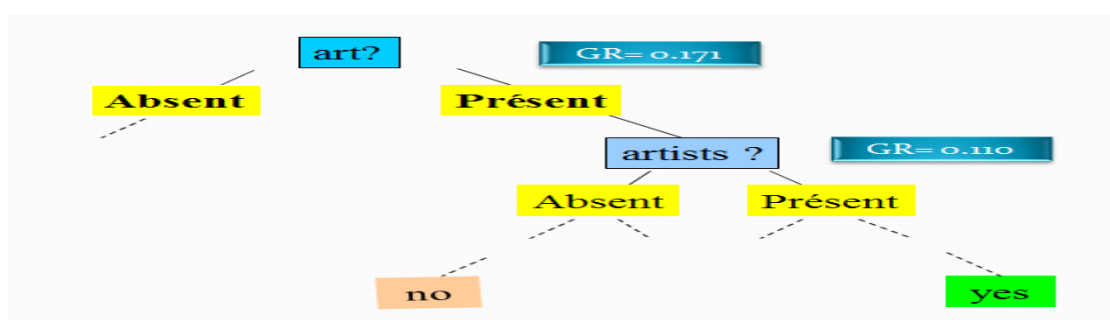


Figure 5.14 : un arbre de décision pour la catégorie art (anglais)

▪ **Langue: Arabe.**Exemple d'expérimentation1 : (Art)

2 fichier de types art.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	02	00
Documents rejets à la catégorie par le classifieur	00	02

Tableau 5.13: Table de contingences pour l'algorithme *arbre de décision* (Art).• **Résultat d'évaluation :**

- Précision : P =1
- Rappel : R =1
- F-mesure =1

Exemple d'expérimentation2 : (économie)

2 fichiers de type économie.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	02	00
Documents rejets à la catégorie par le classifieur	00	02

Tableau 5.14: Table de contingences pour l'algorithme *arbre de décision* (économie).

- **Résultat d'évaluation :**

- **Précision : $P = 1$**
- **Rappel : $R = 1$**
- **F-mesure =1**

Exemple d'expérimentation3 : (politique)

1 fichier de type politique.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	01	00
Documents rejets à la catégorie par le classifieur	02	03

Tableau 5.15: Table de contingences pour l'algorithme *arbre de décision* (politique).

- **Résultat d'évaluation :**

- **Précision : $P = 1$**
- **Rappel : $R = 0,33$**
- **F-mesure =0,50**

Exemple d'expérimentation4 : (sport)

2 fichiers de type sport.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classifieur	01	01
Documents rejets à la catégorie par le classifieur	00	01

Tableau 5.16: Table de contingences pour l'algorithme *arbre de décision* (sport).

- **Résultat d'évaluation :**

- **Précision : $P = 0,5$**
- **Rappel : $R = 1$**
- **F-mesure =0,67**

Taux_Correct = 0,71

Taux_Erreur =0,30

Exemples des modèles d'arbres :

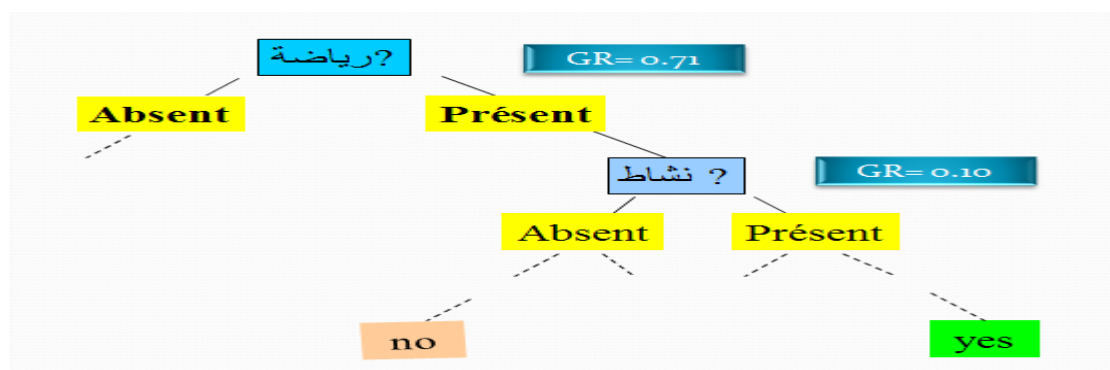


Figure 5.15 : un arbre de décision pour la catégorie sport (arabe)

❖ **Interprétation des résultats obtenus :**

Cette interprétation est confirmée par le taux d'erreur pour l'algorithme arbre de décision « Taux_Erreur », car sa valeur est moins, par contre la valeur de Taux_Correct qui est plus élevée dans tous les cas, c.à.d. que notre algorithme catégorise les documents avec moins d'erreur.

5.9. Conclusion :

Après l'obtention et l'observation des résultats sus-mentionnés, nous pouvons dire que l'objectif fixé au départ et qui consistait à implémenter une application mettant en évidence le processus de classification pour des textes en français ,anglais et arabe, à bien été atteint [26].