Chapitre 03 :

La technique d'authentification du Saint Coran (texte Coranique) proposée

1. Introduction :

Dans ce chapitre, nous présentons une technique d’authentification du Saint Coran sous forme d’un texte.
Nous avons choisi le texte coranique à cause de la majorité des utilisateurs qui utilisent le texte coranique.
Dans ce chapitre, nous allons spécifier l'authentification du Saint Coran qui fonctionne hors ligne en utilisant le texte du Coran (Hafs) comme une base de données, et utiliser des bibliothèques dans le traitement du langage naturel.

2. Stratégie proposée :

Tout d'abord, nous allons montrer la stratégie proposée et en la parler de manière générale.
Nous pouvons montrer la stratégie comme on le voit à la figure 3.1 et l'expliquer en 5 parties.

- L’entrer de texte coranique qui nous avons besoin de l’authentifier.
- L’entrer de texte coranique (le texte de Saint Coran complet dans notre BDD).
- Le prétraitement des deux textes précédents.
- L’algorithme d’indexation.
- Le résultat après le traitement global.

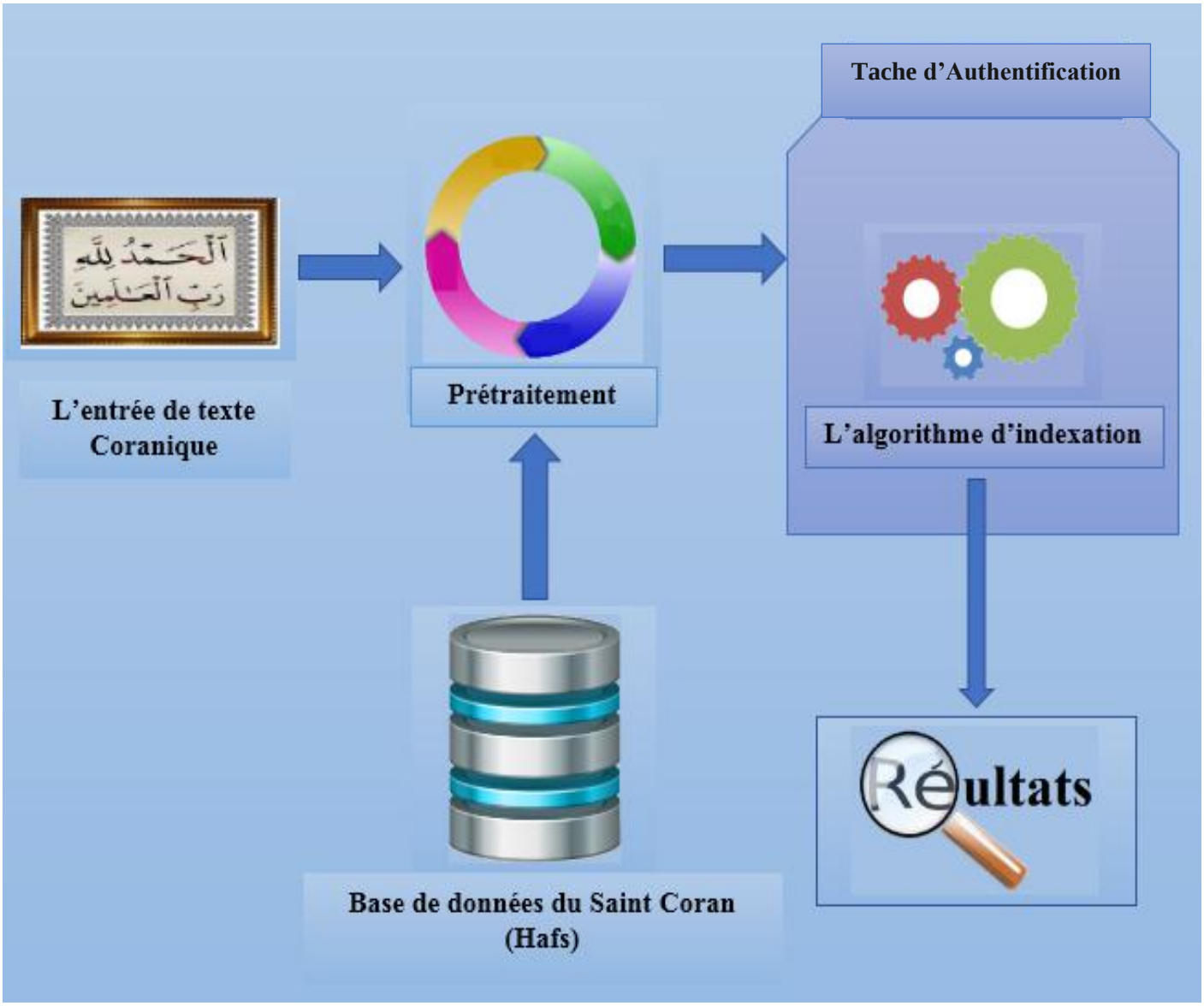


Figure 3.1 : La stratégie proposée

Comme on voit la figure, elle explique les 5 parties que nous avons proposées pour authentifier le texte coranique.

après cela, nous allons spécifier ces parties par en détails.

2.1 Prétraitement + L'entrer des textes :

Voici la figure 3.2 qui explique le prétraitement :

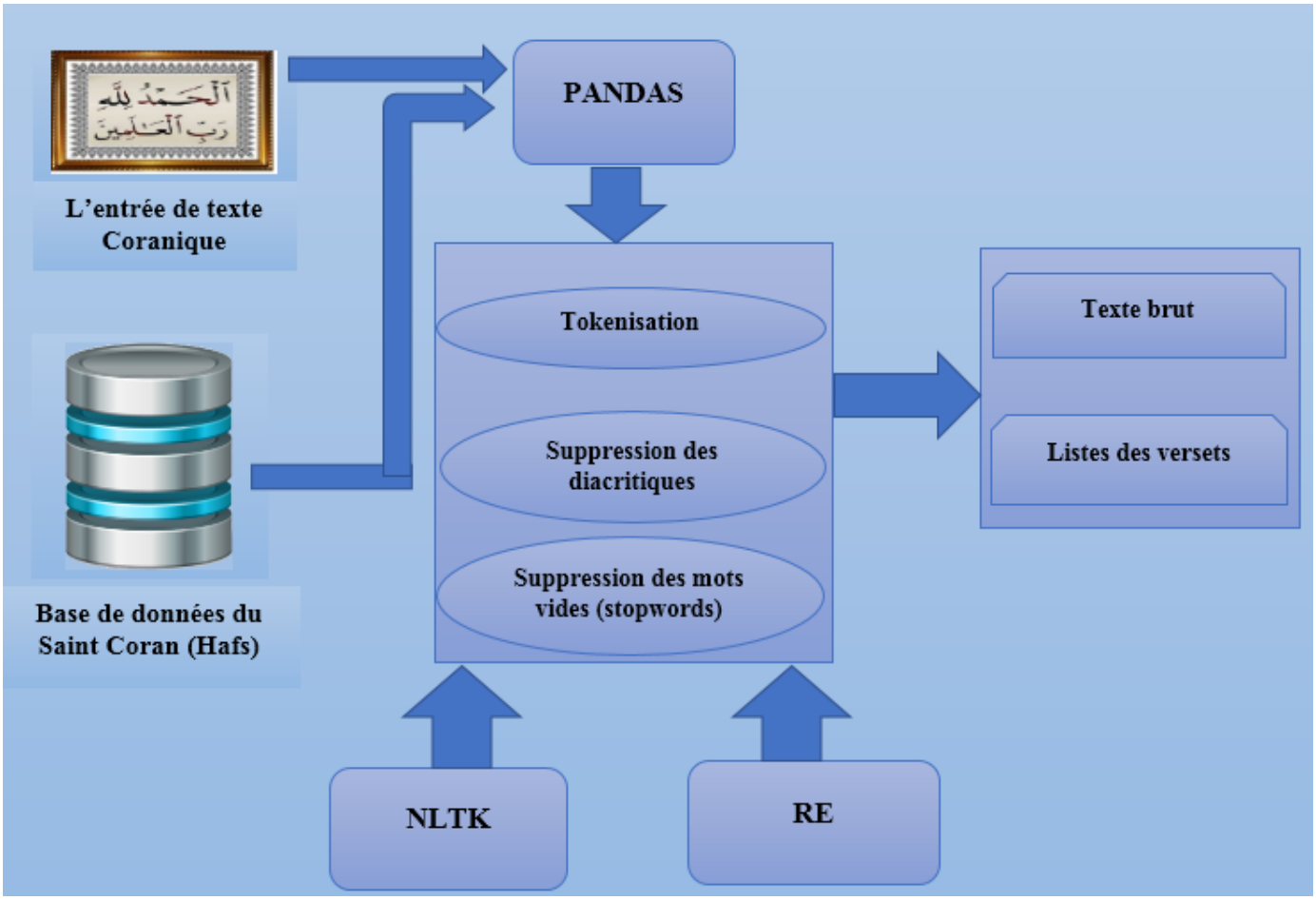


Figure 3.2 : L'explication de prétraitement.

Avant le prétraitement, nous avons le texte d'entrée et le texte coranique que nous avons dans la base de données.

Au cours de l'explication de cette partie, nous allons mentionner les bibliothèques que nous avons utilisées avec ses buts.

- La bibliothèque PANDAS extrait et analyse chacun des textes saisis et le texte coranique de la base de données et les amène à poursuivre le traitement.
- Maintenant, chacun des textes d'entrée et du texte coranique dans la base de données et les listes de versets dans le prétraitement.
- La bibliothèque NLTK a tokenisée les textes dans le prétraitement (séparez le texte en lettres).
- La bibliothèque RE (expressions régulières) supprime les signes diacritiques et les mots vides (stopwords).
- Après le prétraitement, chacun des textes saisis et du texte Coranique dans la base de données sera tokenisés et rédigés sous forme de texte simple.
- Enfin les textes simples (sans signes diacritiques et sans mots vides) sont prêts à passer à l'étape suivante dans la tâche d'Authentification.

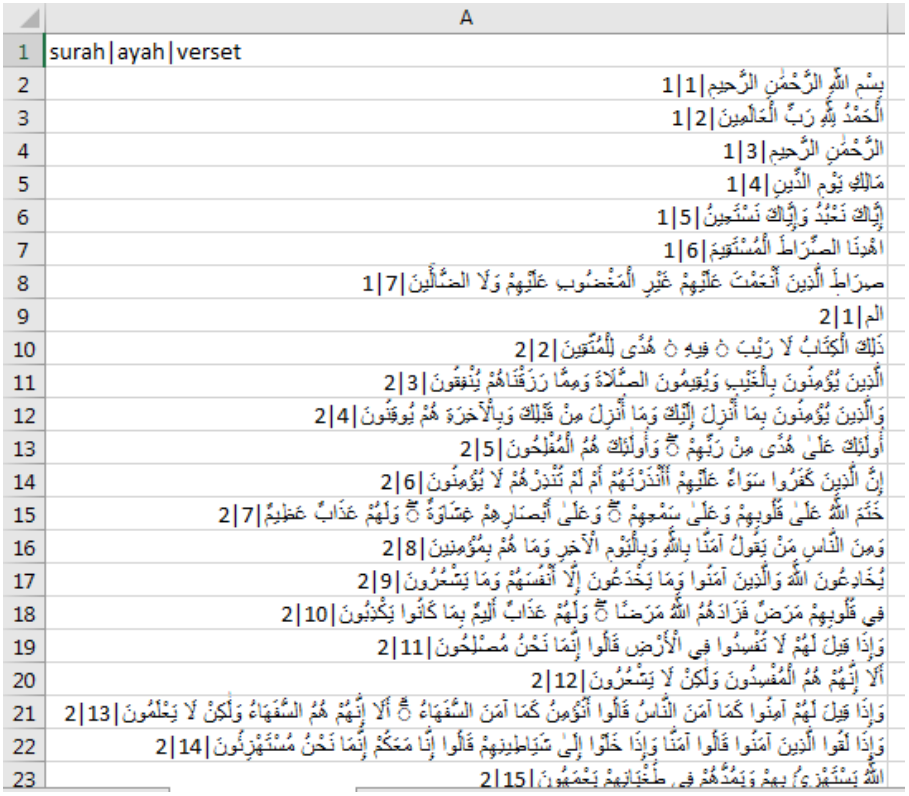
Voici des parties des programmations qui expliquent ses étapes.

```
#EXTRACTION DES BIBLOITHEQUES DE NOTRE TECHNIQUE
>>> import pandas as pd
>>> import nltk
>>> import re
```

Figure 3.3 : Extrait des bibliothèques dans la partie de prétraitement.

```
#Charger le Coran de CSV dans un cadre de données
>>> df = pd.read_csv('data/arabic-original.csv', sep='|', header='infer');
#Supprimez les signes diacritiques des versets et des mots vides pour
simplifier le corpus
>>> df['verse'] = df['verse'].map(lambda x: re.sub('[ٖٗ]', '', x))
#Tokeniser les mots de versets
>>> df['verse'] =df['verse'].str.split()
```

Figure 3.4 : Un exemple de ces bibliothèques comment marchent.



	A
1	surah|ayah|verset
2	بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ|1|1
3	الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ|1|2
4	الرَّحْمَٰنِ الرَّحِيمِ|1|3
5	مَالِكِ يَوْمِ الدِّينِ|1|4
6	إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ|1|5
7	اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ|1|6
8	صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ|1|7
9	الم|2|1
10	ذَٰلِكَ الْكِتَابُ لَا رَيْبَ ۛ فِيهِ ۛ هُدًى لِلْمُتَّقِينَ|2|2
11	الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنْفِقُونَ|2|3
12	وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنْزِلَ إِلَيْكَ وَمَا أُنْزِلَ مِنْ قَبْلِكَ وَبِالْآخِرَةِ هُمْ يُوقِنُونَ|2|4
13	أُولَٰئِكَ عَلَىٰ هُدًى مِنْ رَبِّهِمْ ۖ وَأُولَٰئِكَ هُمُ الْمُفْلِحُونَ|2|5
14	إِنَّ الَّذِينَ كَفَرُوا سَوَاءٌ عَلَيْهِمْ أَأَنْذَرْتَهُمْ أَمْ لَمْ تُنْذِرْهُمْ لَا يُؤْمِنُونَ|2|6
15	خَتَمَ اللَّهُ عَلَىٰ قُلُوبِهِمْ وَعَلَىٰ سَمْعِهِمْ ۖ وَعَلَىٰ أَبْصَارِهِمْ غِشَاوَةٌ ۖ وَلَهُمْ عَذَابٌ عَظِيمٌ|2|7
16	وَمِنَ النَّاسِ مَنْ يَقُولُ آمَنَّا بِاللَّهِ وَبِالْيَوْمِ الْآخِرِ وَمَا هُمْ بِمُؤْمِنِينَ|2|8
17	يُخَادِعُونَ اللَّهَ وَالَّذِينَ آمَنُوا وَمَا يَخْدَعُونَ إِلَّا أَنْفُسَهُمْ وَمَا يَشْعُرُونَ|2|9
18	فِي قُلُوبِهِمْ مَرَضٌ فَزَادَهُمُ اللَّهُ مَرَضًا ۖ وَلَهُمْ عَذَابٌ أَلِيمٌ بِمَا كَانُوا يَكْذِبُونَ|2|10
19	وَإِذَا قِيلَ لَهُمْ لَا تُفْسِدُوا فِي الْأَرْضِ قَالُوا إِنَّمَا نَحْنُ مُصْلِحُونَ|2|11
20	أَلَا إِنَّهُمْ هُمُ الْمُفْسِدُونَ وَلَٰكِنْ لَا يَشْعُرُونَ|2|12
21	وَإِذَا قِيلَ لَهُمْ آمِنُوا كَمَا آمَنَ النَّاسُ قَالُوا أَنُؤْمِنُ كَمَا آمَنَ السُّفَهَاءُ ۗ أَلَا إِنَّهُمْ هُمُ السُّفَهَاءُ وَلَٰكِنْ لَا يَعْلَمُونَ|2|13
22	وَإِذَا لَقُوا الَّذِينَ آمَنُوا قَالُوا آمَنَّا وَإِذَا خَلَوْا إِلَىٰ شَيَاطِينِهِمْ قَالُوا إِنَّا مَعَكُمْ إِنَّمَا نَحْنُ مُسْتَهْزِئُونَ|2|14
23	اللَّهُ يَسْتَهْزِئُ بِهِمْ وَيَمُدُّهُمْ فِي طُغْيَانِهِمْ يَعْمَهُونَ|2|15

Figure 3.5 : Le texte de Saint Coran (Hafs) sous forme CSV.



	A	B
1	surah|	
2		
3	الفاتحة	
4	البقرة	
5	آل عمران	
6	النساء	
7	المائدة	
8	الأنعام	
9	الأعراف	
10	الأنفال	
11	التوبة	
12	يونس	
13	هود	
14	يوسف	
15	الرعد	
16	إبراهيم	
17	الحجر	
18	النحل	
19	الإسراء	
20	الكهف	
21	مريم	
22	طه	
23	الأنبياء	

surah

Figure 3.6 : Listes des titres de chaque chapitre sous forme CSV.

Voici un exemple de prétraitement :

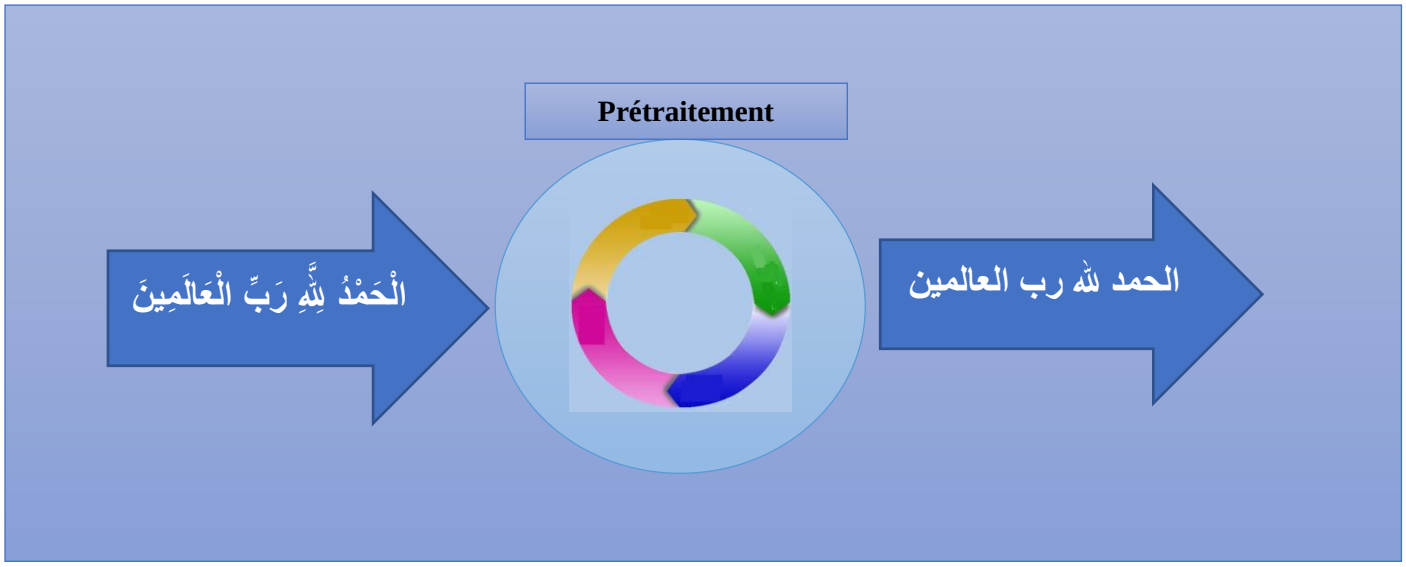


Figure 3.7 : Exemple de prétraitement.

2.2 Tache d’authentification :

Dans la tâche d'authentification, il existe un algorithme d'indexation.
Nous allons spécifier cet algorithme dans 4 parties :

- Filtrage des versets
- Rechercher des versets dans la base de données similaires aux versets de texte d’entré
- L’indexation des versets similaire
- La sélection des versets

Voici la figure 3.8 qui explique cette tâche :

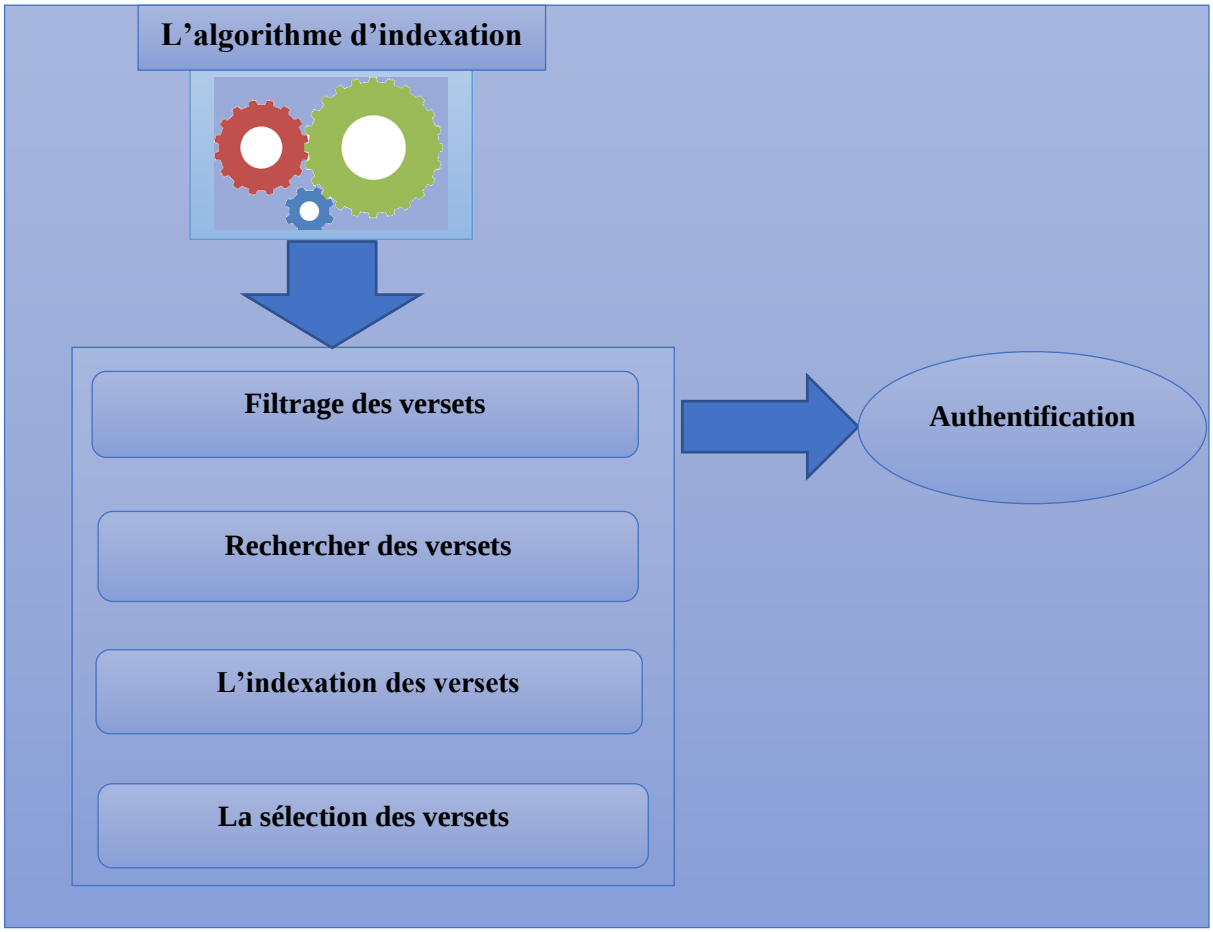


Figure 3.8 : La tache d'authentification**.**

Les textes sous forme de jetons entrent dans l'algorithme d'indexation en deux parties.
L'algorithme filtrera les versets des deux textes (texte d'entrée et texte du Coran).
Après cela, l'algorithme cherchera dans le (texte du Coran) s'il y a une similitude avec le texte de saisi.
L'algorithme indexera les versets de similarité.
Enfin, l'algorithme sélectionnera les vers et affichera le résultat (authentification du texte avec les numéros et le titre du chapitre).Si l'indexation n'est pas vraie à 100%, l'algorithme s'arrête et affiche le résultat (le texte n'est pas authentifié).

Si par exemple nous avons deux versets coraniques dans l'entrée non organisés, l'algorithme les mettra comme non authentifiés même s'ils sont authentifiés séparément.
Voici la figure 3.9 qui montre notre exemple :

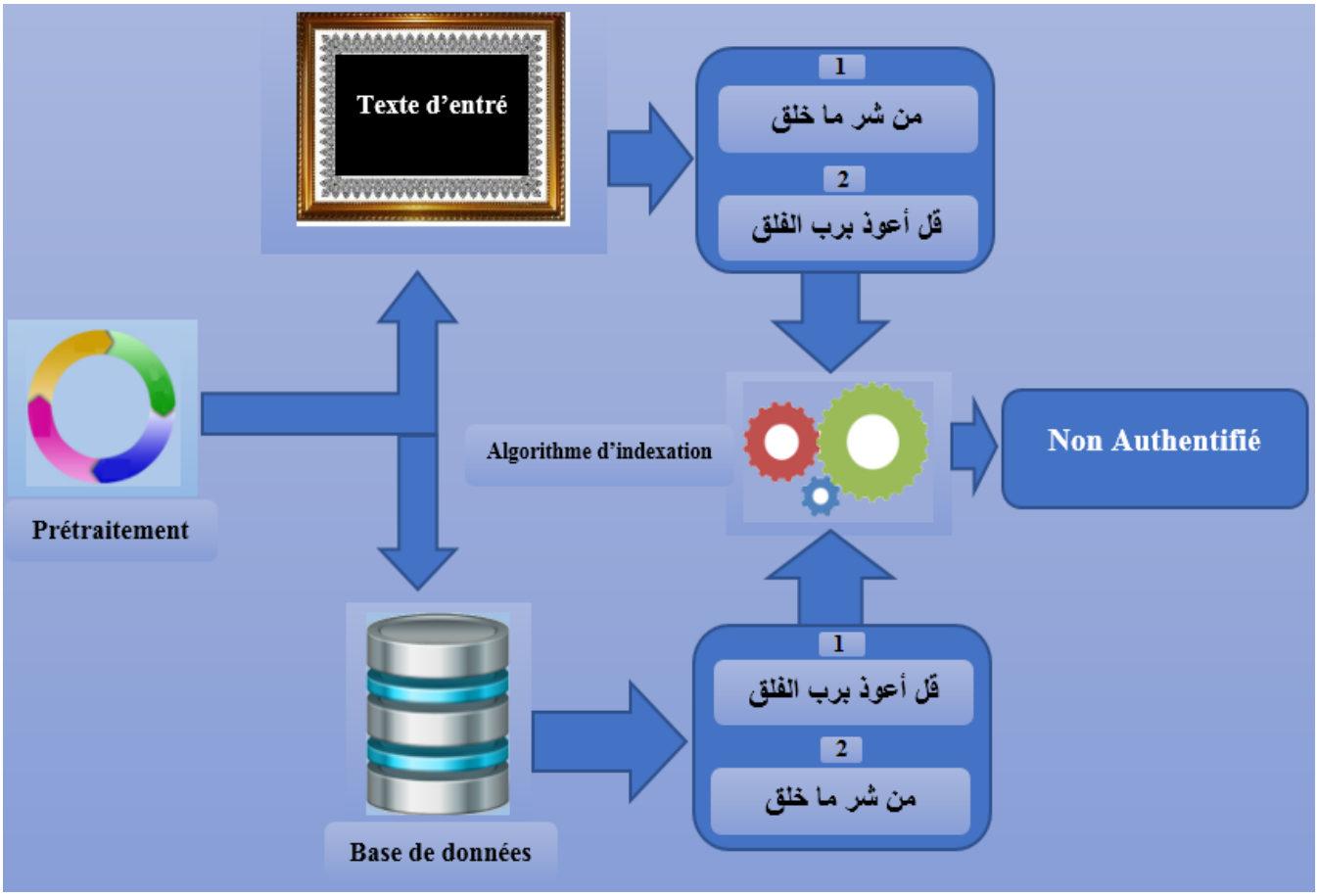


Figure 3.9 : Exemple de l'ordre des versets de texte d'entré

Comme on le voit dans cet exemple, après le prétraitement, nous avons les deux textes (textes simples).
Mais l'ordre des versets n'est pas le même, donc après l'algorithme d'indexation il affichera que le texte d'entré n'est pas authentifié.

Alors l'ordre des versets est très important dans l'authentification du Saint Coran.

- **Remarque :**

Comme nous savons que l'écriture arabe commence de droite à gauche (inverse de l'écriture latine), nous devons donc inverser toute l'écriture arabe pour obtenir une lecture parfaite.

Voici la figure 3.10 qui explique notre travail :

```
# Inverser le texte d'arabe
def reverse(self,s):
 return s[::-1]
# Charger le Coran de CSV dans un cadre de données
df = pd.read_csv('data/arabic-original.csv', sep='|', header='infer');
```

Figure 3.10 : Explication de l'inverse des textes Coraniques

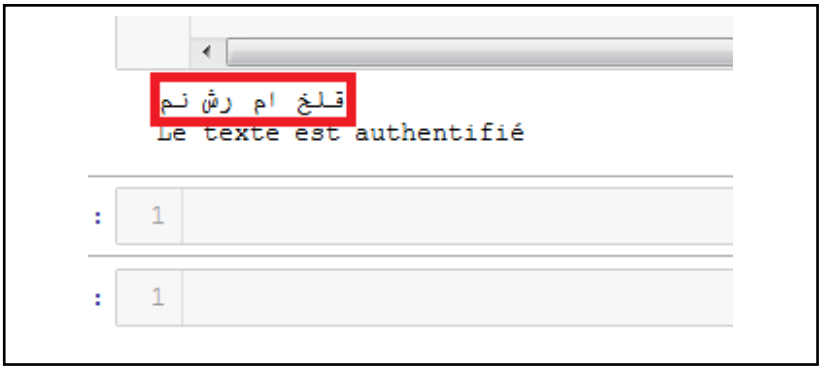


Figure 3.11 : Un exemple d'affichage sans inversion

Comme on le voit sur la figure 3.11, l'écriture arabe sans inversion, se lisant de gauche à droite.

cette fonction est pour tous les utilisateurs, car certains n'ont pas besoin d'inverser le texte arabe et d'autres ont besoin a cause de l'installation de texte d'arabe dans la machine.

2.2.1 Explication d'algorithme d'indexation (des parties de programmation)

Nous allons montrer des parties de programmation pour expliquer l'algorithme comment ça marche

2.2.1.1 L'extraction des versets selon leurs chapitres :

```
    # -----------------extraire  les versus selon leur chapitre----------
     surah = pd.read_csv('data/surah.csv', sep='|', header='infer');
     for j in range(len(a)):
         x=x+"verse "+str(a[j])+" surah "+str(surah['surah'][s[j]])+" \n"
     self.entre1.insert('0.0',x)

 def excute(self):
     vers=[]
     aya=[]
     soura=[]
     df = pd.read_csv('data/arabic-original.csv', sep='|', header='infer');
```

Figure 3.12 : Extraction des versets selon leurs chapitres

2.2.1.2 Filtrage des versets :

```
#------------------------ filtrage--------------
if i+1 != len(df['verse']) and w in self.tostring(df['verse'][i])+''+self.tostring(df['verse'][i+1]) :
        if i+1!=len(df['verse']):
                com.append(df['verse'][i])
                com.append(df['verse'][i+1])
                aya.append(df['verse'][i+1])
                  vers.append(df['ayah'][i])
                  soura.append(df['surah'][i])
                vers.append(df['ayah'][i+1])
                soura.append(df['surah'][i+1])
        else:

            com.append(df['verse'][i])
            aya.append(df['verse'][i])
            vers.append(df['ayah'][i+1])
            soura.append(df['surah'][i+1])

i+=1
```

Figure 3.13 : Filtrage des versets dans les versets sélectionnés

2.2.1.3 Sélectionner les versets :

```
#------------------- selectioner les versets

      if com !=[]:
          print("ok")
          print(aya)
          print(vers)
          print(soura)

      if not(len(w)+1== len(self.tostring(com[1]))):
      if not(len(w)+1== len(self.tostring(com[0]+com[1]))):
      self.variable1.set("Le texte est authentifié mais pas complet")
      self.affiche(vers,soura)
          print("Le texte est authentifié mais pas complet")
      else:
          self.variable1.set("Le texte est authentifié")
          self.affiche([vers[0]],soura)
          print("Le texte est authentifié")
      else:
          self.variable1.set("Le texte est authentifié")
          self.affiche([vers[0]],soura)


      else :
           print("Le texte n'est pas authentifié")
           self.variable1.set("Le texte n'est pas authentifié")

      else :
           print("Entrez votre Texte SVP")
              self.variable1.set("Entrez votre Texte SVP")
```

Figure 3.14 : La sélection des versets

3. Conclusion :

Nous avons présenté dans ce chapitre la stratégie que nous proposons, qui consiste à utiliser la technique d'authentification de texte coranique en utilisant le traitement du langage naturel. nous avons montré l'extraction de textes, en utilisant (pandas) aussi la symbolisation du texte, en utilisant NLTK , et RE pour supprimer les signes diacritiques et les mots vides.
Nous avons montré aussi l'algorithme d'indexation et la sélection des versets à partir des listes.
Dans le chapitre suivant, nous montrerons les résultats de cette technique proposée.