MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS – MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : **Ingénierie des Systèmes d'Information**

Présenté par :
REGUIEG Bochra
DJELLOUL Kenza

THÈME :

Extraction intelligente et interactive des motifs à partir du texte en arabe

Soutenu le : 15 juin 2022

Devant le jury composé de :

Nadia Hocine	MCA	Université de Mostaganem	Présidente
Fatima Zohra Benidris	MCA	Université de Mostaganem	Examinatrice
Sehaba Karim	Professeur	Université de Mostaganem	Encadrant

Année Universitaire 2021-2022

Résumé

Le travail présenté dans ce rapport s'intéresse à l'étude de l'extraction des informations à partir du texte en arabe (plus spécifiquement le sacré coran). La reconnaissance d'entités nommées est une composante essentielle du traitement du langage naturel, permettant l'extraction d'informations et la découverte de connaissances à partir de textes. Notre intérêt porte sur une langue à morphologie complexe, l'arabe, qui présente de grands défis en traitement automatique des langues naturelles. Généralement, les études réalisées concernant l'extraction de l'information à partir du texte ont été développées en anglais et dans certaines langues. Cependant, peu d'études ont été développées en langue arabe. Pour cela, la langue arabe doit effectuer plus de recherches dans ce domaine. Dans le cadre de ce travail, une étude sur l'extraction d'information à partir du texte arabe a été réalisée en se basant sur l'extraction des motifs en utilisant les techniques d'apprentissage machine.

Mots-clés : Extraction d'information (EI), Texte arabe, Extraction des motifs, Apprentissage machine.

Abstract

This report is focusing on the study of the extraction of information from the text in Arabic (more specifically the sacred Quran). Named entity recognition is an essential component of natural language processing, enabling information extraction and knowledge discovery from texts. Our interest is in a language with a complex morphology, Arabic, which presents great challenges in the automatic processing of natural languages. Generally, the studies carried out concerning the extraction of information from the text have been developed in English and in certain languages. However, few studies have been developed in Arabic. For this, the Arabic language needs to do more research in this area. As part of this work, a research study on information extraction from Arabic text was carried out based on pattern extraction using machine learning techniques.

Keywords: Information Extraction (IE), Arabic Text, Pattern Extraction, Machine Learning.

ملخص

يهتم هذا التقرير بدراسة استخراج المعلومات من النص باللغة العربية (وتحديداً القرآن الكريم). يعد التعرف على الكيان المُسمى عنصرًا أساسيًا في معالجة اللغة الطبيعية، مما يتيح استخراج المعلومات واكتشاف المعرفة من النصوص. ينصب اهتمامنا على لغة ذات تركيبة معقدة ، وهي اللغة العربية ، والتي تمثل تحديات كبيرة في المعالجة التلقائية للغات الطبيعية. بشكل عام، تم تطوير الدراسات التي تهتم باستخراج المعلومات من النص باللغة الإنجليزية وبلغات معينة. ومع ذلك ، تم تطوير القليل من الدراسات باللغة العربية. لهذا ، تحتاج اللغة العربية إلى مزيد من البحث في هذا المجال.

كجزء من هذا العمل، تم إجراء دراسة بحثية حول استخراج المعلومات من النص العربي بناءً على استخلاص الأنماط باستخدام تقنيات التعلم الآلي.

الكلمات المفتاحية: استخراج المعلومات، النص العربي ، استخراج الأنماط ، التعلم الآلي .

Dédicaces

Je dédie ce mémoire

A ma chère mère et mon cher père qui n'ont jamais cessé de me soutenir, encouragé durant ces années d'études et de m'épauler pour que je puisse atteindre mes objectifs.

Qu'ils trouvent ici le témoignage de ma profonde reconnaissance.

A mes frères, *Abderrahmane*, *Ismail* et *Mohamed*

A ma chère sœur *Meriem* et son mari *Bayram,* pour ses soutiens morals et leurs conseils précieux. A ma chère petite sœur *Rym,* qui a partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail.

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

A mon cher binôme *Kenza* pour son entente et sa sympathie.

A toutes mes amies *Wassen, Chahinez* et *sarah* qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

A tous ceux que j'aime.

Bochra

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta presence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A mon très cher père

Tu as toujours été à mes côtés pour me soutenir et m'encourager

Que ce travail traduit ma gratitude et mon affection.

A mon très cher frère *Rachid* qui a toujours été là pour moi.

A mes très chères soeurs *Latifa* et *Nihel* et mon beau-frère *Abdou* qui m'ont toujours soutenu et encouragé.

A mes petites nièces adorés *Camélia* et *Farah*.

A mon très cher binôme *Bochra* pour son entente et sa motivation.

A mes très chères cousines *Fazia*, *Hanene*, *Hadjer* qui ont toujours été là pour moi.

Je vous aime tous, puisse Dieu vous donner santé, bonheur, courage et surtout réussite.

Kenza

Remerciements

Je tiens à exprimer toute ma reconnaissance à mon encadrant de mémoire, Monsieur SEHABA Karim. Je le remercie de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Je remercie mes très chers parents, qui ont toujours été là pour moi. Je remercie mes sœurs, et mes frères, pour leurs encouragements.

Je remercie mes amies *Sarah* et *Chahinez* qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Enfin, je remercie ma très chère Kenza qui a contribué au succès de ce mémoire.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Bochra

Je remercie Allah de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche sans l'aide et l'encadrement de Mr Sehaba Karim, je le remercie pour la qualité de son encadrement, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Un grand merci à ma très chère mère et mon très cher père, pour leur amour, leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique, qui m'a permis de réaliser les études que je voulais et par conséquent ce mémoire. Je vous dis merci, parce que ce diplôme, c'est le vôtre.

Je remercie mon frère Rachid et mes sœurs Latifa et Nihel qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Je tiens aussi à adresser mes remerciements à Mr Aimouch pour ses conseils et son soutien inconditionnel.

J'aimerais également remercier mon binôme Bochra qui m'a soutenu et encouragé et qui a participé au succès de ce travail.

Kenza

Liste des figures

Liste des abréviations

Abréviation	Expression Complète
IA	**I**ntelligence **A**rtificielle
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing
DM	**D**ata **M**ining
TM	**T**ext **M**ining
EI	**E**xtraction d'**I**nformations
RI	**R**echerche d'**I**nformations
SW	**S**top **W**ords
LEMs	**L**emmatisation
PS	**P**refix **S**pan
FPG	**F**requent **P**attern **G**rowth

Table des matières

Introduction Générale

Avec le développement des outils informatiques, nous assistons ces dernières années à un accroissement considérable de la quantité d'informations stockées dans de grandes bases de données scientifiques, économiques, financières, médicales, etc. Et le défi aujourd'hui n'est plus de stocker ces données mais d'en extraire de l'information implicite et cachée dans ces données.

La fouille de donnée (DM) et la fouille de donnée textuelle (TM) sont des technologies modernes qui sont utilisées dans le système d'information. Le Text mining (TM) à savoir de l'extraction de l'information utile à partir de gros volumes de contenus textes. Les différentes recherches précédentes de l'extraction d'information à partir des textes conçus beaucoup plus sur des textes français et des textes anglais, les travaux de recherches relatives à l'extraction d'information à partir des textes arabes sont moins nombreux que les autres langues. Parmi les tâches les plus importantes de la fouille de textes (text mining), nous pouvons citer : la recherche d'information, la classification des textes et l'extraction de l'information textuelle dans les textes non structurés (article, blog, message). L'objectif de l'extraction d'information textuelle est de trouver une information précise dans un gros volume de données textuelles, alimenter une base de données, générer un résumé de texte, etc.

D1ans le cadre de ce travail, nous effectuons une poursuite d'un PFE de l'année dernière portant sur la fouille de texte qui consistais à réaliser un système générique d'extraction de motifs fréquents séquentiels et non séquentiels en utilisant des algorithmes d'exploration de données et les appliquer sur un dataset arabe intitulé « Quranic Arabic Corpus », contenant le sacré coran, avec des caractéristiques morphologiques de chaque mot composant ce dernier. Un prétraitement a été appliqué sur le dataset afin de passer à la phase de traitement, en se basant sur l'extraction de la lemmatisation de chaque mot du coran. L'objectif était d'étudier la tâche d'extraction d'information pour la langue arabe, plus précisément dans le corpus du sacré coran.

Et pour se faire, notre démarche est de comprendre le domaine du text mining et son fonctionnement, ainsi de comprendre le travail réalisé l'année dernière et pouvoir poursuivre dans la perspective de mettre évidence certaines limites et pouvoir s'y remédier. C'est dans ce sens que nous avons constaté que l'analyse des résultats d'analyse, relatif au PFE de l'année dernière, n'est pas évidente dans la mesure où l'outil développé ne dispose pas d'une interface graphique facilitant celle-ci.

C’est dans le cadre de cette problématique que nous proposons de développer une interface permettant de gérer les différentes fonctions d’Extraction d’Information à partir du dataset Quranic Arabic Corpus et facilitant l’analyse du texte.

Notre rapport est subdivisé en quartes chapitres. Le premier chapitre intitulé « Etat de l’art sur la fouille de texte » : Dans ce chapitre, nous présentons principalement les notions de fouille de texte et son processus, et quelques définitions relatifs à ce domaine.

Le deuxième chapitre intitulé « Extraction de motifs dans le corpus coranique » : Dans ce chapitre, nous présentons le jeu de données ainsi que les deux principales phases ‘’Prétraitement et Traitement ‘’.

Le troisième chapitre intitulé « Contribution » : Dans ce chapitre, nous présentons l’ensemble de nos contributions concernant le traitement de données, la recherche d’information et la classification.

Le quatrième chapitre intitulé « Discussion des résultats » : Ce dernier chapitre est dédié à la discussion des résultats.

Enfin, nous achèverons notre rapport par une conclusion générale qui résume l’ensemble de nos contributions du présent travail.

Chapitre 1
Etat de l'art sur la fouille de texte

1.1 Introduction

Le texte est l'un des types de données les plus courants dans les bases de données. Selon la base de données, ces données peuvent être organisées comme :

- Données structurées : ces données sont standardisées dans un format tabulaire avec de nombreuses lignes et colonnes, ce qui facilite leur stockage et leur traitement pour les algorithmes d'analyse et d'apprentissage automatique. Les données structurées peuvent inclure des entrées telles que des noms, des adresses et des numéros de téléphone.
- Données non structurées : Ces données n'ont pas de format de données prédéfini. Il peut inclure du texte provenant de sources, telles que des médias sociaux ou des critiques de produits, ou des formats multimédias riches tels que des fichiers vidéo et audio.

L'outil de Text Mining consiste à transformer un texte non structuré en données structurées pour ensuite procéder à l'analyse. Cette pratique repose sur la technologie de « Natural Language Processing » (traitement naturel du langage), permettant aux machines de comprendre et de traiter le langage humain automatiquement.

1.2 Traitement de langage naturel (Natural language Processing)

Le traitement du langage naturel (**NLP**) est une technologie d'intelligence artificielle visant à permettre aux ordinateurs de comprendre le langage humain. **[1]**

1.2.1 Taches de traitement du langage naturel

Dans cette section, nous présentons brièvement les tâches standards du NLP :

- **Part-Of-Speech Tagging :** Est une partie cruciale du traitement du langage naturel. Elle consiste à étiqueter les mots avec une partie du discours, a pour objectif de classer chaque mot avec un signe unique indiquant son rôle syntaxique (à associer à chaque mot d'un texte sa classe morphosyntaxique), par exemple : nom, un verbe, un adjectif…etc. Le POS constitue la base de la résolution d'entité nommée, de l'analyse des sentiments, de la réponse aux questions, et l'ambiguïté du sens des mots.
- **Named Entity Recognition (NER) :** Les étiqueteurs NER (ou reconnaissance d'entités nommées en français) marque les éléments atomiques de la phrase en catégories plus grandes

telles que (noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc.). **[2]**

- **Semantic Role Labeling (SRL) :** SRL vise à donner un rôle sémantique à un constituant syntaxique d'une phrase.
- **Parsing ou Chunking :** Est le processus qui consiste à déterminer la structure syntaxique d'un texte en analysant ses mots constitutifs sur la base d'une grammaire sous-jacente (du langage).

1.2.2 Objectif du NLP

Le domaine du traitement du langage naturel (NLP) vise à convertir le langage humain en une représentation formelle facile à manipuler par les ordinateurs pour étudier des problèmes fondamentaux du traitement de la langue naturelle, ce qui est bien adapté à la modélisation des données textuelles afin d'en extraire des informations et, éventuellement, de représenter les mêmes informations différemment. **[3]**

1.3 Fouille de texte

La Fouille de données textuelles (en anglais appelé Text Mining) est une technique permettant d'automatiser le traitement de gros volumes de contenus texte pour en extraire les principales tendances et répertorier de manière statistique, les différents sujets évoqués ainsi découvrir des connaissances et des relations à partir des documents disponibles. **[4]**

1.3.1 Les principales tâches de fouille de textes

Nous allons énumérer les trois principales tâches auxquelles s'attaque la fouille de textes :**[5]**

- La classification de textes : Elle consiste à ranger des textes ou des documents dans des "classes" prédéfinies.
- La recherche d'informations : Elle est déjà omniprésente dans nos usages quotidiens des ordinateurs. Nous la sollicitons chaque fois que nous recherchons des documents répondant à une "requête".
- L'extraction d'information : Comme son nom l'indique, elle se fixe comme objectif d'*extraire* du texte des informations factuelles précises.

Dans notre recherche, nous nous intéressons à l'Extraction d'Information (EI) à partir du texte qui est une tâche très importante qui permet d'examiner d'importantes collections de documents pour découvrir de nouvelles informations ou aider à répondre à des questions de recherche précises. Bien que cette problématique soit très difficile, les performances des approches proposées dans l'état de l'art s'améliorent car l'intelligence artificielle est désormais capable de classifier automatiquement les textes par sentiment, par sujet ou par intention. Un algorithme de Text Mining est par exemple capable de passer en revue les commentaires sur un produit pour déterminer s'ils sont principalement positifs, neutres ou négatifs. Il est aussi possible de repérer les mots-clés les plus fréquemment employés. L'extraction d'information est un sous-domaine du TLN.

Deux classes de motifs se sont alors avérées très utiles et simultanément utilisées dans la pratique, à savoir :

- Les itemsets fréquents
- Les motifs séquentiels fréquents

1.4 NLP vs Text Mining

NLP fonctionne avec tout produit de la communication humaine naturelle, y compris le texte, la parole, les images, les signes, etc. Il extrait les significations sémantiques et analyse les structures grammaticales que l'utilisateur saisit. Le fouille de textes fonctionne avec des documents textuels. Elle extrait les caractéristiques des documents et utilise une analyse qualitative. NLP fournit la compréhension des sentiments décrits, la structure grammaticale et le sens sémantique.

Ces résultats permettent une traduction fluide du texte vers d'autres langues. L'exploration de texte montre les relations entre les mots du texte.

1.5 Concepts clés de l'extraction des motifs

Dans cette partie nous donnons quelques définitions des concepts clés de l'extraction de motifs :

- **Item :** Est tout objet, article, attribut, littéral, appartenant à un ensemble fini d'éléments distincts I = {x1, x2, ..., x m}. Dans les applications de type analyse du panier de la ménagère, les articles en vente dans un magasin sont des items
- **Motif :** Un motif est un ensemble d'attributs (itemset) dont la valeur doit être vraie.
- **Taille d'un motif :** La taille T d'un motif M est définie comme le cardinal de ses attributs.

- **Support minimal :** Notée minsup(Ii) est le nombre minimum d'occurrence d'un Itemset pour être considéré comme fréquent. L'occurrence n'est prise en compte qu'une fois dans la transaction. C'est un seuil choisi par l'utilisateur.
- **Motif fréquent :** On dira qu'un motif est fréquent si sa fréquence est supérieure à un seuil défini a priori.
- **Motifs maximaux :** Un itemset it_i est dit maximal si it_i est fréquent et s'il n'existe pas d'itemsets fréquents it_j tels que $it_i \subset it_j$. Cette représentation condensée est dite approximative. En effet, elle ne permet pas de calculer précisément le support des itemsets inclus dans le motif maximal. Mais elle permet de dériver une borne inférieure sur le support d'un itemset : le support de chaque motif inclus dans it_i apparait autant ou plus souvent que it_i. Cette représentation condensée s'étend simplement au cadre des motifs séquentiels. **[6]**

1.6 Algorithme d'extraction de motifs

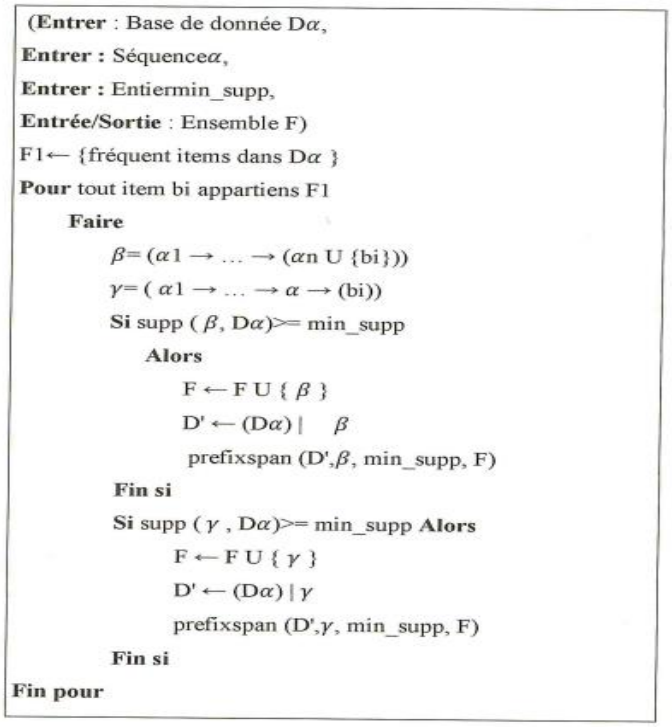
Dans cette section, nous allons présenter quelques algorithmes permettant l'extraction de motifs :

1.6.1 Algorithme PrefixSpan (Prefix-projected Sequential Pattern):

L'algorithme PrefixSpan permet l'extraction de motif séquentiel à l'aide du paradigme motif growth. Cet algorithme utilise la méthode diviser pour régner. Il passe à plusieurs passages qui sont les suivants : **[7]**

- Le premier passage concerne la base de données, permet d'extraire l'ensemble des 1-séquences fréquentes.
- Chaque motif séquentiel est considéré comme un préfixe. L'ensemble complet des motifs séquentiels est ainsi partitionné en différents sous-ensembles par rapport à différents préfixes.
- Extraire les sous-ensembles de motifs séquentiels, des bases de données, projetées sont construites et fouillées récursivement.

1.6.1.1 Pseudo code de l'algorithme PrefixSpan :



```
(Entrer : Base de donnée Dα,
Entrer : Séquenceα,
Entrer : Entiermin_supp,
Entrée/Sortie : Ensemble F)
F1← {fréquent items dans Dα }
Pour tout item bi appartiens F1
    Faire
        β= (α1 → … → (αn U {bi}))
        γ= ( α1 → … → α → (bi))
        Si supp ( β, Dα)>= min_supp
            Alors
                F ← F U { β }
                D' ← (Dα) |   β
                 prefixspan (D',β, min_supp, F)
            Fin si
            Si supp ( γ , Dα)>= min_supp Alors
                F ← F U { γ }
                D' ← (Dα) | γ
                prefixspan (D',γ, min_supp, F)
            Fin si
Fin pour
```

Figure 1 Pseudo code de l'algorithme PrefixSpan

1.6.1.2 Les Avantages de PrefixSpan

L'algorithme PrefixSpan est basé sur schéma de croissance séquentielle (sequence growth pattern). Avec ses avantages en termes de performance et d'efficacité, l'algorithme PrefixSpan est généralement préféré dans le domaine de l'extraction des motifs fréquents. Il présente les avantages suivants:

- Aucune génération de candidats
- La fréquence des articles locaux seulement dénombrables
- La méthode de recherche Divide-and-conquer est utilisée

Il est supérieur à GSP ainsi qu'à FreeSpan. **[8]**

1.6.2 L 'algorithme Apriori

L'algorithme Apriori (Agrawal et Srikant, 1994) représente le premier algorithme de recherche de règle d'association incluant les étapes d'élagage qui permet de tenir en comptes la croissance des items. L'algorithme commence par déterminer le support de chaque item. Puis, il génère les ensembles d'items de taille K à partir des ensembles précédents de taille

(k-l). Il permet de vérifier si l'ensemble d'items est fréquent, alors tous ses sous-ensembles sont aussi fréquents. **[9]**

1.6.2.1 Pseudo code de l'algorithme Apriori :

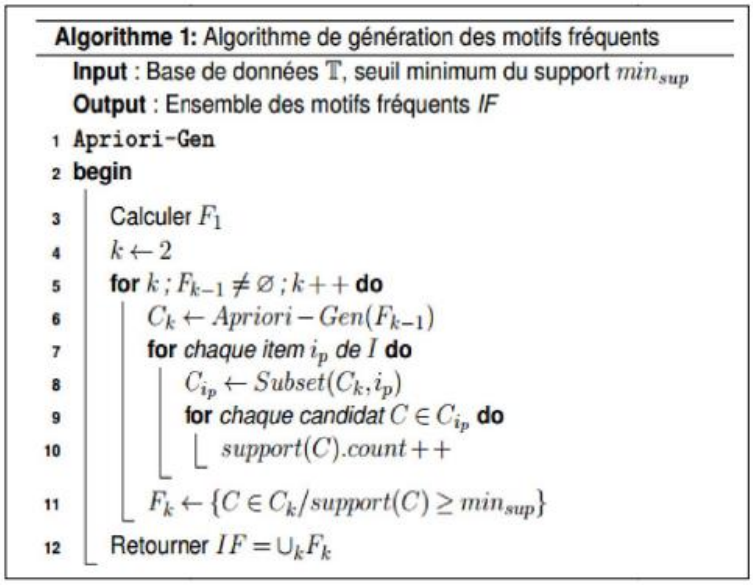


Figure 2: Pseudo code de l'algorithme Apriori

1.6.2.2 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 un balayage de la base de données pour calculer le support de chaque item et la collecte de ces itemsets qui ont un support ≥ support minimum. L'ensemble résultant est noté $L1$ (chaque Lk servit à construire l'étape suivante), puis utilisé pour trouver $L2$, aussi l'ensemble résultant les 2-itemsets est utilisé pour trouver $L3$, et ainsi de suite jusqu'à ce qu'aucun k-itemsets ne puisse être trouvé. L'obtention de chaque Lk nécessite une analyse complète de la base de données (Han et Kamber, 2006). Si un ensemble d'items est fréquent, alors tous ses sous-ensembles sont aussi fréquents.

1.6.2.3 Les Avantages de l'algorithme Apriori

- Facile à comprendre le fonctionnement de l'algorithme.
- Les étapes Join et Prune sont faciles à mettre en œuvre sur de grands itemsets dans de grandes bases de données.

1.6.2.4 Les désavantages de l'algorithme Apriori

- Il nécessite un calcul élevé si les itemsets sont très grands et le support minimum est maintenu très bas.
- Toute la base de données doit être scannée

1.6.3 L'Algorithme Fp-Growth

L'algorithme **Fp-growth** permet la découverte des itemsets fréquents sans génération des itemsets candidats. Le processus se déroule en deux étapes, une étape de construction des arbres FP-tree et une étape d'extraction des itemsets fréquents directement de ces arbres. La construction de l'arbre FP-tree s'effectue suivant les étapes ci-dessous : **[10]**

- Calculer le support minimal.
- Calculer chacune des occurrences d'un item constituant la base de transactions.
- Établir un critère de priorité pour ces items.
- Faire le tri des items en fonction de leur priorité.
- Établir le nœud racine.
- À partir de chaque nœud père insérer les enfants en partant du nœud racine
- Valider la structure de l'arbre FP-Growth.

1.6.3.1 Pseudo code de l'algorithme FP-Growth :

```
procedure FP-Growth (Tree, α)
{
  if    Tree contains a single path P  then
        for each β = nodes combination in P  do
        pattern = β ∪ α;
        support = min(support of the nodes in β);

  else
        for each ai in the header of Tree      do
        pattern β = a i ∪ α;
        with support =  a i. support ;
        construct conditional pattern base of β
        TreeB = construct conditional FP-tree of β
        if      TreeB != Ø     then
        call FP-Growth( TreeB , β)
}
```

Figure 3: Pseudo code de l'algorithme FP-Growth

1.6.3.2 Avantages de FPG

1. Cet algorithme ne doit scanner la base de données que deux fois par rapport à Apriori qui scanne les transactions pour chaque itération.
2. L'appariement des éléments n'est pas fait dans cet algorithme et cela le rend plus rapide.
3. La base de données est stockée dans une version compacte en mémoire.
4. Il est efficace et évolutif pour l'exploitation à la fois des longs et courts motifs fréquents. **[10]**

1.6.3.3 Désavantages de FPG

1. FP Tree est plus lourd et difficile à construire qu'Apriori
2. Couteuse.
3. Lorsque la base de données est grande, l'algorithme peut ne pas tenir dans la mémoire partagée. **[10]**

1.7 Conclusion

Le but de ce premier chapitre était de définir la notion du Text Mining ainsi que son fonctionnement et les concepts clés de l'extraction de motifs, de ce fait nous avons présenté quelques algorithmes permettant l'extraction de motifs.

Chapitre 2
Extraction des motifs dans le corpus coranique

2.1 Introduction

Dans ce chapitre nous présentons une application de statistiques sur le Saint Coran ainsi que le jeu de données sur lequel nous allons appliquer notre analyse. Ensuite, nous présentons les deux principales phases d'analyse '' Prétraitement et Traitement '' sur ce jeu de données.

2.2 Programme de statistiques coraniques : (برنامج الاحصاء القرآني)

Il s'agit d'un logiciel de statistiques du coran, développé par un Algérien, un égyptien et plusieurs chercheurs en sciences du Coran.

Cet outil est entièrement gratuit à télécharger, disponible que sur version Windows. Il est simple et facile à utiliser.**[11]**

Cette version de ce logiciel (4.4.4) a été développée le 13/10/2021.

Voici quelques fonctionnalités du logiciel :

- Recherchez la fréquence d'un mot, d'une lettre, d'une phrase, d'un signe diacritique ou même de points et plus encore.
- Dérivés de mots : recherche de plusieurs dérivés de mots avec plusieurs racines linguistiques.
- Recherche de racines : toutes les statistiques liées aux versets, sourates, mots et lettres
- Agencement de la recherche : La possibilité de rechercher selon l'ordre mecquois et médinois, ou par les versets les plus longs et les plus courts et l'ordre inverse... Il est également possible d'afficher un texte spécifique du Coran pour en découvrir facilement l'explication de ses versets.
- Copie et exportation : Les textes coraniques sont gratuits et disponibles pour copie dans un fichier Word ou Excel.
- Recherche par numéro : Recherche par numéro de lettre... numéro de mot... numéro de verset... au niveau d'un verset, d'une sourate ou du Coran entier.

- Connaître l'ordre de n'importe quel mot, lettre ou verset depuis le début ou la fin du Coran ou à l'intérieur d'une sourate ou d'un groupe de versets ou d'une sourate.

L'interface d'accueil :

Comme nous pouvons le voir, en ouvrant le programme, l'interface d'accueil apparaît avec le détail des versets : le numéro de la sourate, le numéro du verset, le nombre de lettres avec leurs détails et le nombre de mots.

Figure 4: L'interface d'accueil du programme

Nous allons présenter quelques fonctionnalités intéressantes que le programme offre :

- La fonctionnalité « Statistiques avec racines » que nous allons identifier ses caractéristiques avec l'exemple suivant :

  Nous effectuons notre recherche sur le LEM "أله" ainsi que ses dérivés linguistiques, et le programme affichera tous les mots qui ont la racine "أله". Comme le montre la figure suivante :

Figure 5: Capture d'une recherche sur le LEM أله et ses dérivés linguistiques

- Recherche de deux racines ou plus par ordre séquentiel / non séquentiel :

  Par exemple nous effectuons la recherche sur les deux racines رحم -غفر et tous les versets contenant ces deux racines vont apparaître.
  Aussi nous pouvons changer les choix de recherche, par exemple rechercher les versets qui contiennent les deux racines ensemble ou bien au moins une des deux racines par ordre séquentiel ou non séquentiel.

Figure 6: Capture sur une recherche de versets contenant les deux racines par ordre séquentiel

Figure 7: Capture sur une recherche de versets contenant les deux racines par ordre non séquentiel

Nous constatons que ce logiciel est basé sur une analyse statistique auquel l'utilisateur doit introduire des mots clés. Cette approche est différente de la nôtre dans la mesure où notre objectif est de détecter des motifs fréquents en se basant sur la taille du motif et sa fréquence.

Ainsi, l'utilisateur n'aura pas à saisir des mots clés mais de renseigner ces deux paramètres. Afin de réaliser notre objectif, nous nous sommes basés sur le jeu de données « Quranic Corpus Arabic » présenté ci-après.

2.3 Présentation du jeu de données « Quranic Corpus Arabic »

Corpus coranique arabe, une ressource linguistique annotée qui montre la grammaire, la syntaxe et la morphologie arabes de chaque mot du Saint Coran. Le corpus propose trois niveaux d'analyse : l'annotation morphologique, une treebank syntaxique et une ontologie sémantique. Une treebank est une ressource linguistique qui rassemble des arbres syntaxiques. Ce sont des analyses de phrases annotées manuellement qui peuvent être lues à la fois par les humains et les ordinateurs, avec des treebanks différents adoptant différentes théories de syntaxe. La plus récente recherche informatique en langue arabe se concentre sur l'arabe standard moderne, et l'arabe classique du Coran a été relativement inexploré. Presque aucune attention n'a été accordée à la grammaire arabe traditionnelle, malgré de nombreux volumes écrits sur le sujet au cours des siècles. **[12]**

La distribution Uthmani du projet **Tanzil** est utilisée (http://tanzil.info) et n'est pas modifiée. C'est une représentation exacte de la Madina Mushaf. Le texte est stocké sous forme de document XML Unicode, avec un élément XML pour chaque chapitre et verset du Coran.

Il n'est pas facile de trouver un système de POS disponible, robuste et précis pour traiter les textes coraniques en arabe, car il s'agit d'un texte sacré et d'une langue dont la structure morphologique est complexe. Pour décrire le corpus coranique pour tous les lecteurs, le corpus coranique arabe (Ducs et Habash, 2010) est une ressource linguistique intégrée et fiable qui se compose de 77430 mots d'arabe coranique, divisé en 114 documents. Chaque mot est étiqueté avec sa partie de la parole ainsi que de multiples caractéristiques morphologiques qui sont basées sur le traditionnel Grammaire arabe. En outre, il est stocké dans un fichier texte et est disponible gratuitement. Les données dans le corpus est écrit dans le schéma de translittération arabe de Buckwalter. **[12]**

2.3.1 Buckwalter

La translittération de Buckwalter utilise des caractères ASCII pour représenter l'orthographe arabe. Comme il y a une correspondance un-à-un avec Unicode, le schéma d'encodage est réversible. **[12]**

Translittération étendue de Buckwalter : Il y a 4 caractères non-arabiques dans la sorcière d'encodage originale ne se trouvent pas dans le texte coranique : P (peh), J (tcheh), V (veh) et G (gaf). Le caractère de combinaison alif + maddah (|) n'est pas non plus utilisé dans Tanzil XML. Ces caractères ne sont pas implémentés par l'encodeur JQuranTree Buckwalter. De même, 14 symboles coraniques ne figurent pas dans le schéma original. Dans le schéma étendu utilisé par JQuranTree, ceux-ci ont été attribués aux marques de ponctuation ASCII. Ce n'est pas ambigu, car la ponctuation moderne ne se produit pas dans le Coran :

- Maddah (^)
- Hamza Above (#)
- Small High Seen (:)
- Small High Rounded Z ero (@)
- Small HighU pright Rectangular Zero (")
- Small High Meem Isolated Form ([)
- Small Low Seen (;)
- Small Waw (,)
- Small Ya (.)
- Small High Noon (!)
- Empty Centre Low Stop (-)
- Empty Centre High Stop (+)

- Rounded High Stop With Filled Centre (%)
- Small Low Meem (])

Le schéma étendu de translittération de Buckwalter est illustré dans la figure 9. ci-dessous.

Les sections surlignées en jaune indiquent les parties ajoutées par rapport à l'original : **[12]**

UNICODE			BUCKWALTER	
Decimal	Hex	Glyph	ASCII	Orthography
1619	U+0653	ـٓ	^	Maddah
1620	U+0654	ـٔ	#	HamzaAbove
1648	U+0670	ـٰ	`	AlifKhanjareeya
1649	U+0671	ٱ	{	Alif + HamzatWasl
1756	U+06DC	ۜ	:	SmallHighSeen
1759	U+06DF	۟	@	SmallHighRoundedZero
1760	U+06E0	۠	"	SmallHighUprightRectangularZero
1762	U+06E2	ۡ	[	SmallHighMeemIsolatedForm
1763	U+06E3	ۣ	;	SmallLowSeen
1765	U+06E5	ۥ	,	SmallWaw
1766	U+06E6	ۦ	.	SmallYa
1768	U+06E8	ۨ	!	SmallHighNoon
1770	U+06EA	۪	-	EmptyCentreLowStop
1771	U+06EB	۫	+	EmptyCentreHighStop
1772	U+06EC	۬	%	RoundedHighStopWithFilledCentre
1773	U+06ED	ۭ	]	SmallLowMeem

Figure 8: Les parties ajoutées dans le schéma étendu de translittération de Buckwalter

2.3.2 Composant de jeu de donnes

Le jeu de données est organisé en quatre colonnes comme suit :

1. LOCATION : comprend quatre parties :
   - 1.1.N° chapitre : c'est-à-dire en arabe (رقم السورة)
   - 1.2.N° verset : c'est-à-dire en arabe (رقم الاية)
   - 1.3.N° mot : c'est-à-dire la position d'un mot dans un verset par exemple :

     bi=1 , somi=1 , {ll~ahi=2.
   - 1.4.N°partie : c'est-à-dire le numéro des parties d'un mot par exemple :

le Préfixe bi = 1 et la racine somi = 2.

2. FORM : comprend les parties principales du mot . **[12]**

3. TAG : comprend l'étiquette (POS) de la partie du discours pour chaque partie du mot
tels que Nom, Verbe, Adjectif, etc,

Nous avons remarqué qu'il est écrit sous la forme suivante : POS : la valeur de Part of speech et cette valeur peux être l'un de ces tags comme montre le tableau suivant : **[12]**

	Tag	Arabic Name	Description
Nouns	N	اسم	Noun
	PN	اسم علم	Proper noun
Derived nominals	ADJ	صفة	Adjective
	IMPN	اسم فعل أمر	Imperative verbal noun
Pronouns	PRON	ضمير	Personal pronoun
	DEM	اسم اشارة	Demonstrative pronoun
	REL	اسم موصول	Relative pronoun
Adverbs	T	ظرف زمان	Time adverb
	LOC	ظرف مكان	Location adverb

Figure 9: Part of speech de la partie de discours pour les nominaux

4. FEATURES : décrit les caractéristiques morphologiques du mot, notamment : Racine, LMAs, sexe, etc. Nous pouvons mentionner quelques caractéristiques :

Préfixe : En plus de part of speech de la parole, de multiples fonctions d'inflexion sont assignées à chaque segment morphologique. Par exemple, les caractéristiques pour la personne, le sexe et le nombre. Les caractéristiques pour les préfixes se terminent en + et sont montrées dans les figures ci-dessous. En revanche, les suffixes commencent par +. **[12]**

Feature	Name	Segment part-of-speech / description
Al+	determiner (*al*)	DET – determiner prefix ("the")
bi+	preposition (*bi*)	P – preposition prefix ("by", "with", "in")
ka+	preposition (*ka*)	P – preposition prefix ("like" or "thus")
ta+	preposition (*ta*)	P – particle of oath prefix used as a preposition ("by Allah")
sa+	future particle (*sa*)	P – prefixed particle indicating the future ("they will")
ya+	vocative particle (*yā*)	VOC – a vocative prefix usually translated as "O"
ha+	vocative particle (*hā*)	VOC – a vocative prefix usually translated as "Lo!"

Figure 10: Caractéristiques identifiant les segments préfixés

Feature	Name	Segment part-of-speech / description
A:INTG+	interrogative particle (*alif*)	INTG – prefixed interrogative particle ("is?", "did?", "do?")
A:EQ+	equalization particle (*alif*)	EQ – prefixed equalization particle ("whether")

Figure 11: Caractéristiques identifiant la particule alif comme préfixe

Feature	Name	Segment part-of-speech / description
w:CONJ+	conjunction (*wa*)	CONJ – conjunction prefix ("and")
w:REM+	resumption (*wa*)	REM – resumption prefix ("then" or "so")
w:CIRC+	circumstantial (*wa*)	CIRC – circumstantial prefix ("while")
w:SUP+	supplemental (*wa*)	SUP – supplemental prefix ("then" or "so")
w:P+	preposition (*wa*)	P – particle of oath prefix used as a preposition ("by the pen")
w:COM+	comitative (*wa*)	COM – comitative prefix ("with")

Figure 12: Caractéristiques identifiant la particule wāw comme préfixe

Feature	Name	Segment part-of-speech / description
l:P+	preposition (*lām*)	P – the letter *lām* as a prefixed preposition
l:EMPH+	emphasis (*lām*)	P – the letter *lām* as a prefixed particle used to give emphasis
l:PRP+	purpose (*lām*)	P – the letter *lām* as a prefixed particle used to indicate purpose
l:IMPV+	imperative (*lām*)	P – the letter *lām* as a prefixed particle used to form an imperative

Figure 15: Caractéristiques identifiant la particule lām comme préfixe

Feature	Name	Segment part-of-speech / description
f:REM+	resumption (*fa*)	REM – resumption prefix ("then" or "so")
f:CONJ+	conjunction (*fa*)	CONJ – conjunction prefix ("and")
f:RSLT+	result (*fa*)	RSLT – result prefix ("then")
f:SUP+	supplemental (*fa*)	SUP – supplemental prefix ("then" or "so")
f:CAUS+	cause (*fa*)	CAUS – cause prefix ("then" or "so")

Figure 14: Caractéristiques identifiant la particule fa comme préfixe

Feature	Name	Description
ROOT:	root	Indicates the (usually triliteral) root of a word, for example ROOT:ktb
LEM:	lemma	Specifies the common lemma for a group of words, for example LEM:kitaAb
SP:	special	Indicates that the word belongs to a special group, for example SP:<in~

Figure 13 : Caractéristiques des racines et des LEMs

Racine et LEMs : En arabe et d'autres langues sémitiques comme l'hébreu, des mots semblables peuvent être regroupés selon une racine. Il s'agit d'une séquence de 3 ou 4 consonnes (appelées radicaux) qui forment ensemble une racine trilitère ou quadrilatérale. A partir d'une seule racine, une grande variété de mots peuvent être formés, avec des significations distinctes mais connexes. Par exemple à partir de la racine trilittérale kāf tā bā (ك ت ب) le verbe "écrire" peut-être formé, ainsi que ses dérivés en arabe y compris "écrire", "livre", "auteur", "bibliothèque" et "bureau". Le concept de lemme est également utilisé pour regrouper des mots similaires à un niveau de granularité plus fin qu'une racine. Les lemmes regroupent des formes-mots qui ne diffèrent que par leur morphologie ineffective (par opposition à dérivée) et ne varient pas dans leur signification. Contrairement à la racine, le lemme est un mot réel sélectionné pour représenter le groupe et est généralement le même mot que celui utilisé dans les titres du dictionnaire. Une troisième caractéristique utilisée pour regrouper les mots est la fonction SP (spéciale). Certains groupes de verbes et de particules ont des règles particulières en grammaire arabe en ce qui concerne les terminaisons de cas et les rôles syntaxiques. **[12]**

- **Personne, sexe et nombre** : En arabe, les mots peuvent infléchir pour la personne, le sexe et le nombre. Contrairement aux mots anglais inflexion non seulement pour le pluriel et le singulier, mais aussi pour le double. Par exemple, il y a un mot-forme distinct pour représenter "deux livres". Dans le corpus coranique arabe, les caractéristiques de la personne, du sexe et du nombre sont combinées à l'aide d'une notation concaténâtes. Par exemple, 3MS représente la troisième personne, masculine, singulière. De même, 2D représente la deuxième personne, double. Le concept de genre dans la grammaire arabe peut faire référence au genre sémantique, morphinique ou grammatical (voir la grammaire du genre). **[12]**

Feature	Arabic Name	Values	Description
person	الاسناد	1, 2, 3	first person, second person, third person
gender	الجنس	M, F	masculine, feminine
number	العدد	S, D, P	singular, dual, plural

Figure 16: Caractéristiques des Personne, sexe et nombre

D'où nous pouvons conclure qu'ils y ont des caractéristiques différentes, séparé part '|' comme montre la figure suivante :

```
LOCATION          FORM      TAG       FEATURE
(112:1:1:1)       qulo      V
          STEM|POS:V|IMPV|LEM:qaAla|ROOT:qwl|2MS
(112:1:2:1)       huwa      PRON
          STEM|POS:PRON|3MS
(112:1:3:1)       {ll~ahu   PN
          STEM|POS:PN|LEM:{ll~ah|ROOT:Alh|NOM
(112:1:4:1)       >aHadN    N
          STEM|POS:N|LEM:>aHad|ROOT:AHd|M|INDEF|N
OM
```

Figure 17 : Un verset d'Alikhlas en schéma de translittération de Buckwalter.

2.4 Prétraitement de texte

Nous allons citer les différentes étapes du prétraitement utilisés : **[13]**

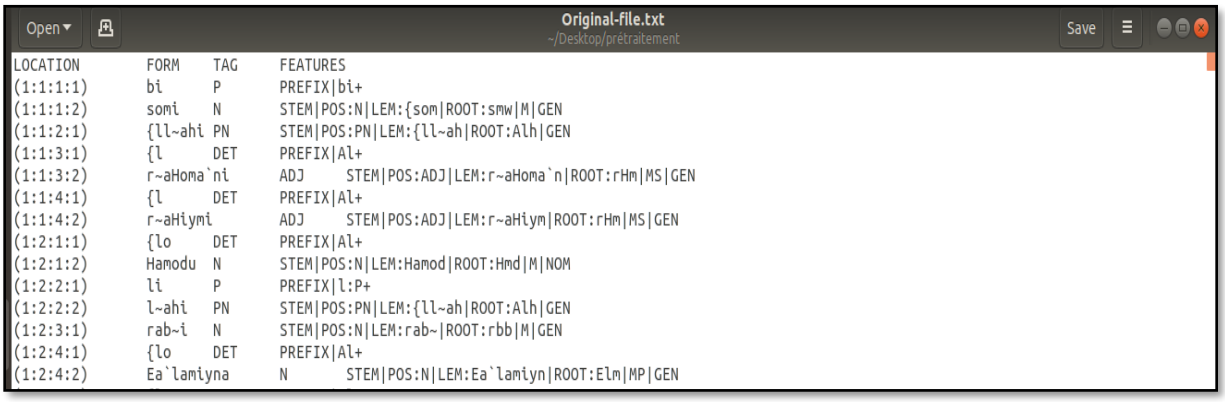


Figure 18: Le fichier texte original du dataset quranic arabic corpus

- **Etape1 :** Consiste à prendre le fichier original du dataset pour découper la colonne FEATURES en plusieurs sous-colonnes et remplacer le '|' par une tabulation. Cette première étape a pour but de faciliter la prise des LEMs.

A	B	C	D	E	F	G	H
LOCATION	FORM	TAG	FEATURE				
(1:1:1:1)	bi	P	PREFIX	bi+			
(1:1:1:2)	somi	N	STEM	POS:N	LEM:{som	ROOT:smw	M
(1:1:2:1)	{ll~ahi	PN	STEM	POS:PN	LEM:{ll~ah	ROOT:Alh	GEN
(1:1:3:1)	{l	DET	PREFIX	Al+			
(1:1:3:2)	r~aHoma`ni	ADJ	STEM	POS:ADJ	LEM:r~aHoma`n	ROOT:rHm	MS
(1:1:4:1)	{l	DET	PREFIX	Al+			
(1:1:4:2)	r~aHiymi	ADJ	STEM	POS:ADJ	LEM:r~aHiym	ROOT:rHm	MS
(1:2:1:1)	{lo	DET	PREFIX	Al+			

Figure 19: Résultat de l'étape 1 de la phase prétraitement de données

- **Etape2 :** Utilise le fichier de l'étape 1 (avec des features séparés) et renomme les colonnes après la décomposition de la colonne FEATURES par Feature-1,… Feature-10.

A	B	C	D	E	F	G	H
LOCATION	FORM	TAG	Feature-1	Feature-2	Feature-3	Feature-4	Feature-5
(1:1:1:1)	bi	P	PREFIX	bi+			
(1:1:1:2)	somi	N	STEM	POS:N	LEM:{som	ROOT:smw	M
(1:1:2:1)	{ll~ahi	PN	STEM	POS:PN	LEM:{ll~ah	ROOT:Alh	GEN
(1:1:3:1)	{l	DET	PREFIX	Al+			
(1:1:3:2)	r~aHoma`ni	ADJ	STEM	POS:ADJ	LEM:r~aHoma`n	ROOT:rHm	MS
(1:1:4:1)	{l	DET	PREFIX	Al+			
(1:1:4:2)	r~aHiymi	ADJ	STEM	POS:ADJ	LEM:r~aHiym	ROOT:rHm	MS

Figure 20: Résultat de l'étape 2 de la phase prétraitement de données

- **Etape3 :** Cette étape consiste à garder que les lignes contenant les LEMs, tout en supprimant les préfixes, suffixes…, elle utilise comme entrée le résultat de l'étape précédente.

A	B	C	D	E	F	G	H
(1:1:1:2)	somi	N	STEM	POS:N	LEM:{som	ROOT:smw	M
(1:1:2:1)	{ll~ahi	PN	STEM	POS:PN	LEM:{ll~ah	ROOT:Alh	GEN
(1:1:3:2)	r~aHoma`ni	ADJ	STEM	POS:ADJ	LEM:r~aHoma`n	ROOT:rHm	MS
(1:1:4:2)	r~aHiymi	ADJ	STEM	POS:ADJ	LEM:r~aHiym	ROOT:rHm	MS

Figure 21 : Résultat de l'étape 3 de la phase prétraitement de données

- **Etape4 :** Cette étape contient deux fonctions, la fonction keep_only_LEM_elements(file) : fait l'extraction de toutes les cellules contenant LEM :'valeur'. Où le paramètre « file » est le fichier contenant que les lignes qui ont LEM :'valeur' dans l'une de ses cellules.

La fonction splitting_LEMAS_column : utilise le résultat de la fonction précédente (la colonne contenant LEM :'valeur') et va prendre que la valeur du LEM.

Résultat de keep_only_LEM_elements

LEM:{ll~ah
LEM:r~aHoma`n
LEM:r~aHiym
LEM:Hamod
LEM:{ll~ah
LEM:rab~
LEM:Ea`lamiyn
LEM:r~aHoma`n
LEM:r~aHiym
LEM:ma`lik
LEM:yawom
LEM:diyn

Résultat de splitting_LEMAs_column

{ll~ah
r~aHoma`n
r~aHiym
Hamod
{ll~ah
rab~
Ea`lamiyn
r~aHoma`n
r~aHiym
ma`lik
yawom
diyn

Figure 22: Résultat de l'étape 4 de la phase prétraitement de données

- **Etape5 :** Cette étape va faire l'extraction du numéro de chapitre et de verset de la colonne LOCATION en utilisant le fichier qui contient les lignes avec LEM :'valeur'. Cette étape facilite la concaténation des LEMs avec leur LOCATION dans l'étape suivante.

	A	B	C
1	N-chapter	N-verse	
2	1	1	
3	1	1	
4	1	1	
5	1	1	
6	1	2	
7	1	2	
8	1	2	
9	1	2	
10	1	3	
11	1	3	

Figure 23: Résultat de l'étape 5 de la phase prétraitement de données

Etape6 : Cette étape consiste à faire la concaténation des LEMs avec leur emplacement (N-chapitre, N-verset).

	A	B	C	D
1	N-chapter	N-verse	LEMAs	
2	1	1	{som	
3	1	1	{ll~ah	
4	1	1	r~aHoma`n	
5	1	1	r~aHiym	
6	1	2	Hamod	
7	1	2	{ll~ah	
8	1	2	rab~	
9	1	2	Ea`lamiyn	
10	1	3	r~aHoma`n	

Figure 24: Résultat de l'étape 6 de la phase prétraitement de données

- **Etape7 :** Cette étape fait la suppression des Stop-words (min, man,..). Et pour se faire il faut d'abord connaitre la fréquence de chaque mot pour ensuite traduire le résultat de buckwalter vers l'arabe et le comparer avec celle traduite en arabe et enfin les mettre dans un fichier StoWords.txt dans une liste séparée par des '|'.

	A	B	C
1	word	frequency	
2	min	3226	
3	{ll~ah	2699	
4	maA	2565	
5	laA	1738	
6	fiY	1701	
7	<in~	1682	
8	qaAla	1618	
9	{l~a*iY	1464	
10	EalaY`	1445	
11	kaAna	1358	
12	rab~	975	
13	man	871	
14	<ilaY`	742	
15	<in	697	
16	<il~aA	663	
17	>an	625	

	A	B
1	word	frequency
2	مِن	3226
3	ٱللَّه	2699
4	مَا	2565
5	لَا	1738
6	فِى	1701
7	إِنَّ	1682
8	قَالَ	1618
9	ٱلَّذِى	1464
10	عَلَىٰ	1445
11	كَانَ	1358
12	رَبّ	975
13	مَن	871
14	إِلَىٰ	742
15	إِن	697
16	إِلَّا	663
17	أَن	625

Figure 25: Résultat de l'étape 7 de la phase prétraitement de données

- **Etape8 :** Cette étape consiste à mettre chaque verset dans une ligne en utilisant le fichier contenant l'emplacement et les LEMs sans les Stop-Words.

A	B	
N-chapter	N-verse	LEMAs
1	1	{som|{ll~ah|r~aHoma`n|r~aHiym
1	2	Hamod|{ll~ah|rab~|Ea`lamiyn
1	3	r~aHoma`n|r~aHiym
1	4	ma`lik|yawom|diyn
1	5	<iy~aA|Eabada|<iy~aA|{sotaEiynu
1	6	hadaY|Sira`T|m~usotaqiym
1	7	Sira`T|>anoEama|gayor|magoDuwb|DaA^l~
2	2	kita`b|rayob|hudFY|mut~aqiyn
2	3	'aAmana|gayob|>aqaAma|Salaw`p|razaqa|>anfaqa
2	4	'aAmana|>anzala|>anzala|A^xir|yuwqinu
2	5	>uwla`^}ik|hudFY|rab~|>uwla`^}ik|mufoliHuwn

Figure 27: Résultat de l'étape 8 de la phase prétraitement de données

- **Etape9 :** Cette étape, va extraire que la liste des versets en utilisant le fichier qui contient chaque verset dans une ligne avec leur emplacement.

LEMAs
{som|{ll~ah|r~aHoma`n|r~aHiym
Hamod|{ll~ah|rab~|Ea`lamiyn
r~aHoma`n|r~aHiym
ma`lik|yawom|diyn
<iy~aA|Eabada|<iy~aA|{sotaEiynu
hadaY|Sira`T|m~usotaqiym

Figure 26: Résultat de l'étape 9 de la phase prétraitement de données

- **Etape10 :** Cette dernière étape, consiste à séparer les versets collés par des '|' en utilisant le fichier contenant chaque verset dans une ligne. Et ça sera l'étape qui va afficher le fichier final prétraité.

	A	B	C	D	E	F	G
1	{som	{ll~ah	r~aHoma`n	r~aHiym			
2	Hamod	{ll~ah	rab~	Ea`lamiyn			
3	r~aHoma`n	r~aHiym					
4	ma`lik	yawom	diyn				
5	<iy~aA	Eabada	<iy~aA	{sotaEiynu			
6	hadaY	Sira`T	m~usotaqiym				
7	Sira`T	>anoEama	magoDuwb	DaA^l~			
8	kita`b	rayob	hudFY	mut~aqiyn			
9	'aAmana	gayob	>aqaAma	Salaw`p	razaqa	>anfaqa	
10	'aAmana	>anzala	>anzala	A^xir	yuwqinu		
11	>uwla`^}ik	hudFY	rab~	>uwla`^}ik	mufoliHuwn		
12	kafara	sawaA^'	>an*ara	>an*ara	'aAmana		
13	xatama	{ll~ah	qalob	samoE	baSar	gi$a`wap	Ea*aAb

Figure 28: Résultat de l'étape 10 de la phase prétraitement de données

2.5 Discussion

La méthode d'analyse des résultats relative au PFE de l'année dernière, n'est pas évidente et est considérée comme une tâche lourde et difficile à effectuer dans la mesure où l'outil développé ne dispose pas d'une interface graphique facilitant celle-ci. Et donc, pour modifier les paramètres on est obligé de passer par le code source et fixer les paramètres des 3 algorithmes : le support minimum, la longueur minimale et maximale du motif et la fréquence. Aussi le fichier de résultat est affiché au format CSV et l'analyse est faite manuellement en faisant une transformation en fichier Excel et en appliquant les filtres pour ensuite faire la comparaison. Ce qui rend le travail complexe et nécessite obligatoirement une interface permettant de réaliser les différentes tâches en un seul clic.

Pour cela, nous proposons de développer une interface permettant de faciliter les tâches en permettant à l'utilisateur de choisir l'algorithme souhaitant travailler avec, ainsi de faire entrer les paramètres des algorithmes. Le système affiche le résultat et l'utilisateur pourra filtrer les données en choisissant par exemple d'afficher les motifs par ordre décroissant de fréquence, de taille…etc. Le système pourra ainsi donner la possibilité à l'utilisateur d'afficher plusieurs résultats.

2.6 Conclusion

Dans ce chapitre, nous avons présenté le jeu de données « Corpus Arabe Coranique » utilisé dans notre étude ainsi que ses trois différents niveaux d'analyses. De plus nous avons cité les principales démarches de fouille de textes : le prétraitement du texte et le traitement du texte.

Chapitre 3
Contribution

3.1 Introduction

Dans ce chapitre nous allons présenter notre démarche en détaillant le processus de traitement de données, la recherche d'information et la classification, puis nous allons présenter l'environnement matériel et logiciel utilisé, enfin nous présenterons notre système d'extraction de motifs.

3.2 Démarche

3.2.1 Traitement de données

Dans cette phase de traitement, un test du jeu de données prétraité (résultat de la phase de prétraitement) a été appliqué sur les 3 algorithmes précédemment présentés :

Test avec PrefixSpan : le test a été effectué sur l'algorithme prefixSpan qui sert à extraire les motifs séquentiels fréquents. Cet algorithme va prendre comme paramètres : le fichier prétraité, le support minimum d'itemsets, la longueur minimale et maximale du motif.

Et on aura comme résultat les motifs séquentiels fréquents avec leurs fréquences et leurs tailles.

Test avec FP-Growth : le test a été effectué sur l'algorithme FP-Growth qui est un algorithme d'extraction de motifs fréquents. Il prend comme paramètres : le fichier prétraité, le support minimum, la longueur minimale et maximale du motif. Et comme résultat les motifs non séquentiels fréquents avec leurs fréquences et leurs tailles.

Test avec Apriori : le test a été effectué sur l'algorithme Apriori, qui est comme le FP-Growth un algorithme d'extraction de motifs fréquents. Cet algorithme qui prend comme paramètres : le fichier prétraité, le support, le nombre de transactions (le nombre de versets), la longueur minimale et maximale du motif. Le support minimum sera calculé comme suit : support/N-transaction. Et on aura comme résultat les motifs non séquentiels fréquents avec leurs fréquences et leurs tailles.

3.2.2 Recherche d'information

Après avoir lancé l'exécution d'un des algorithmes, l'utilisateur peut effectuer une recherche par mots et aura comme résultat tous les motifs contentant le(s) mot(s) recherché(s) en prenant en compte les paramètres de l'algorithme introduits (support, taille minimale, taille maximale). Sinon, le résultat obtenu sera celui de l'exécution de l'algorithme sans prise en compte mots à rechercher.

En cliquant sur l'un des résultats, ça nous permet d'afficher tous les versets écrits en arabe contenant ce motif.

3.2.3 Classification

La classification des données est le processus consistant à analyser des données structurées ou non structurées et à les organiser en classes.

Dans notre cas, la classification nous permet de classer les motifs fréquents qui apparaitront dans les versets selon leur similarité, en parcourant tous les motifs du résultat de la recherche.

Tout d'abord, nous calculons la similarité en utilisant la fonction de similarité « Jaccard » de tous les motifs entre eux. S'il y'a une similarité supérieure ou égale au seuil introduit par l'utilisateur, nous enchaînons l'étape suivante qui permet d'extraire le(s) mot(s) commun(s) des motifs similaires. Si aucune similarité n'a été détectée, la liste des motifs sera retournée telle quelle.

En premier temps, nous affichons le résultat sous la forme suivante :

'Mot(s) commun(s)' : 'la classe contenant les motifs ayant le même mot commun'

Sinon, on descend récursivement vers un autre niveau si possible jusqu'à ce que la liste des motifs soit vide et par la suite, nous affichons la hiérarchie de chaque classe et on aura un graphe pour chaque hiérarchie.

3.2.3.1 Similarité de Jaccard

L'indice de Jaccard (Jaccard, 1901), permet de mesurer les similitudes entre les ensembles. Il est défini par la taille de l'intersection notée $|A \cap B|$ divisée par la taille de l'union $|A \cup B|$, pour deux classes quelconques A et B. Cet indice se calcule comme suit : **[14]**

$$S_J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

3.3 Implémentation

3.3.1 Environnement matériel et logiciel

La réalisation de notre système d'extraction de motifs fréquents, a été faite dans l'environnement matériel et logiciel suivant :

- CPU : Intel® Core™ i3-4005U @ 1.7 GHz.
- RAM : 4 GO.
- Système d'exploitation : Windows 7, 64-bit.

Langages et outils de développement : Nous avons utilisé le langage de programmation Python 3, utilisant l'IDE Visual Studio Code et la bibliothèque graphique Tkinter de création d'interfaces graphiques.

Python 3 : est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Visual Studio Code : est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions.

Tkinter : (Tk interface) est un module intégré à la bibliothèque standard de Python, permettant de créer des interfaces graphiques

3.3.2 Processus de traitement

3.3.2.1 Algorithme de Prefix Span

Prefix span est un algorithme d'extraction des motifs séquentiels fréquents, le code suivant montre l'application de ce dernier sur notre dataset : il prend comme paramètre :

- file_in : Le fichier csv prétraité de notre jeu de données.

- Min_support : La fréquence minimale des motifs.
- Min_pattern_length : La longueur minimale de motifs que nous voulons avoir.
- Max_pattern_length : : La longueur maximale de motifs que nous voulons avoir.
- File out : c'est le lien de fichier csv ou nous allons sauvegarder le résultat.

Résultat d'utilisation de ce code avec :

- min_support=30
- min_pattern_length=3
- max_pattern_length=10

Résultat du test :

Freq	Motif	Taille
30	ra'aA,{ll~ah,{ll~ah	3
30	samaA^',>aroD,$aYo'	3
30	>aroD,{ll~ah,{ll~ah	3
30	qaAla,qaAla,rab~	3
30	{ll~ah,n~aAs,{ll~ah	3
30	{ll~ah,{ll~ah,hadaY	3
31	'aAlaA^',rab~,ka*~aba	3
31	>aY~,'aAlaA^',ka*~aba	3
31	>aY~,'aAlaA^',rab~,ka*~aba	4
31	>aY~,rab~,ka*~aba	3

Figure 29 : Résultat du teste avec l'algorithme Prefix Span

- **Conversion des motifs de buckwalter en arabe :**

Nous avons ajouté une fonction permettant de convertir les versets contenant les motifs des résultats précèdent de buckwalter en arabe, prenant un exemple le motif 'aAmana,Eamila,S~a`liHa`t

Et avec les paramètres : Support=30, Taille_min=1 et Taille_max=10.

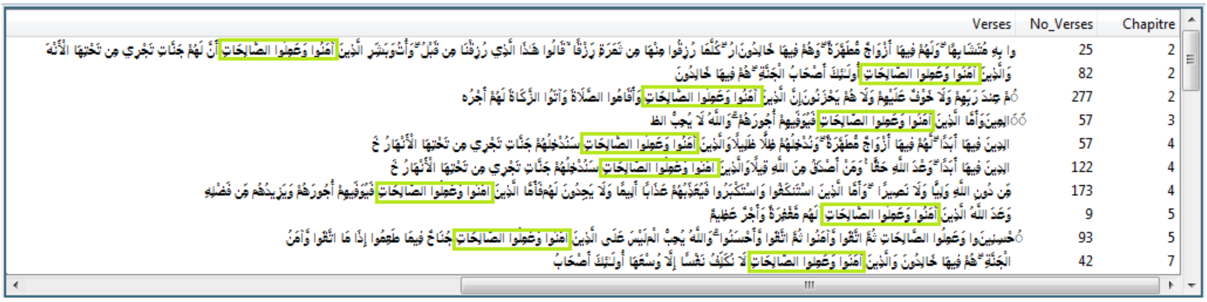


Figure 30: résultat des versets convertis en arabe avec le motif déjà recherché

3.3.2.2 Algorithme de FP-growth

FP-growth est un algorithme d'extraction de motifs fréquents, le code suivant montre l'application de ce dernier sur notre dataset : il prend comme paramètre :

- file_in : Le fichier csv prétraité de notre jeu de données.
- Min_support : La fréquence minimale des motifs.
- Min_pattern_length : La longueur minimale de motifs que nous voulons avoir.
- Max_pattern_length : : La longueur maximale de motifs que nous voulons avoir.
- File out : c'est le lien de fichier csv ou nous allons sauvegarder le résultat.

Si nous testons ce code avec avec :

- min_support=20.
- min_pattern_length=3.
- max_pattern_length=30.

Résultat du test :

Fréquen	Motif	Taille
42	rab~,qaAla,qawom	3
28	kaAna,qaAla,qawom	3
30	kaAna,qawom,{ll~ah	3
50	qaAla,qawom,{ll~ah	3
23	duwn,{t~axa*a,{ll~ah	3
20	qaAla,{t~axa*a,{ll~ah	3
20	rab~,qaAla,muwsaY`	3
20	qaAla,muwsaY`,{ll~ah	3
20	{bon,qaAla,{ll~ah	3
21	Salaw`p,>aqaAma,A^taY,zakaw`p	4

Figure 31 : Résultat du test avec l'algorithme Fp-Growth

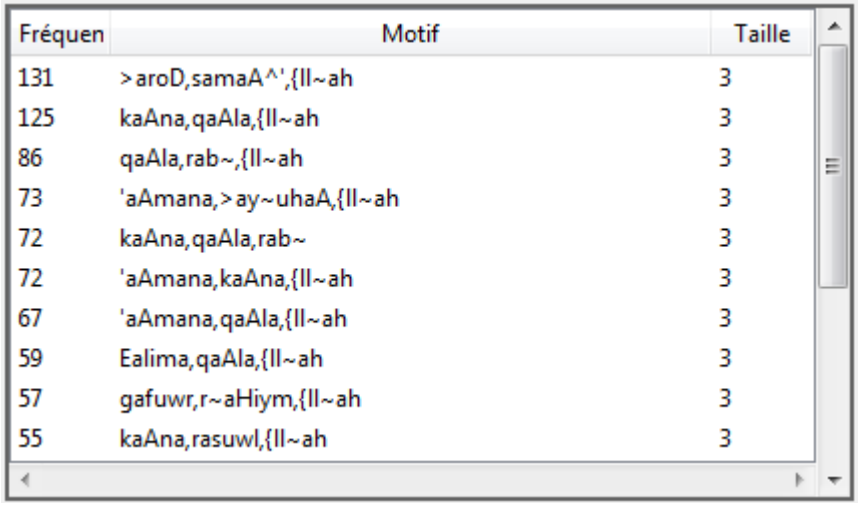
3.3.2.3 Algorithme d'Apriori

Apriori est un algorithme d'extraction des motifs séquentiels fréquents, le code suivant montre l'application de ce dernier sur notre dataset : il prend comme paramètre :

- file_in : Le fichier csv prétraité de notre jeu de données.
- Min_support : La fréquence minimale des motifs.
- Min_pattern_length : La longueur minimale de motifs que nous voulons avoir.
- Max_pattern_length : : La longueur maximale de motifs que nous voulons avoir.
- File out : c'est le lien de fichier csv ou nous allons sauvegarder le résultat.
- N-transaction : c'est le nombre total de transaction, dans notre cas c'est : 6216, c'est-à-dire le nombre de versets.

Si nous teston cet algorithme avec

- Support=50
- N_transaction =6216.
- min_pattern_length=3.
- max_pattern_length=20
- Le min_support dans l'algorithme sera calculé par 50/6216=0.008



Fréquen	Motif	Taille
131	>aroD,samaA^',{II~ah	3
125	kaAna,qaAla,{II~ah	3
86	qaAla,rab~,{II~ah	3
73	'aAmana,>ay~uhaA,{II~ah	3
72	kaAna,qaAla,rab~	3
72	'aAmana,kaAna,{II~ah	3
67	'aAmana,qaAla,{II~ah	3
59	Ealima,qaAla,{II~ah	3
57	gafuwr,r~aHiym,{II~ah	3
55	kaAna,rasuwl,{II~ah	3

Figure 32: Résultat de traitement de notre jeu de données avec l'algorithme d'apriori

3.3.3 Description de l'interface développée

Dans cette section, nous présentons notre système d'extraction de motifs séquentiels et non séquentiels à partir du saint coran et ses fonctionnalités.

Nous avons choisi de nommer notre application AYAT, et ça signifie Versets en arabe (ايات).

Les fonctionnalités de notre application :

- Rechercher les motifs fréquents selon un support/fréquence et une taille (de la séquence) introduits par l'utilisateur.
- Choisir un algorithme spécifique pour faire la recherche.
- Organiser le résultat de recherche par ordre croissant/décroissant de fréquence, taille…
- Possibilité de faire la recherche avec 2 ou 3 algorithmes différents et d'afficher les 2 ou 3 résultats sur la même page pour que l'analyste puisse comparer.
- Rechercher toutes les séquences contenant un ou plusieurs mots saisis par l'utilisateur.

- Afficher le résultat de recherche d'une séquence sous-forme de graphe hiérarchique : la partie supérieure de graphe contient les mots les plus fréquents/communs et la partie inférieur les mots les moins fréquents/communs.

Interface d'accueil

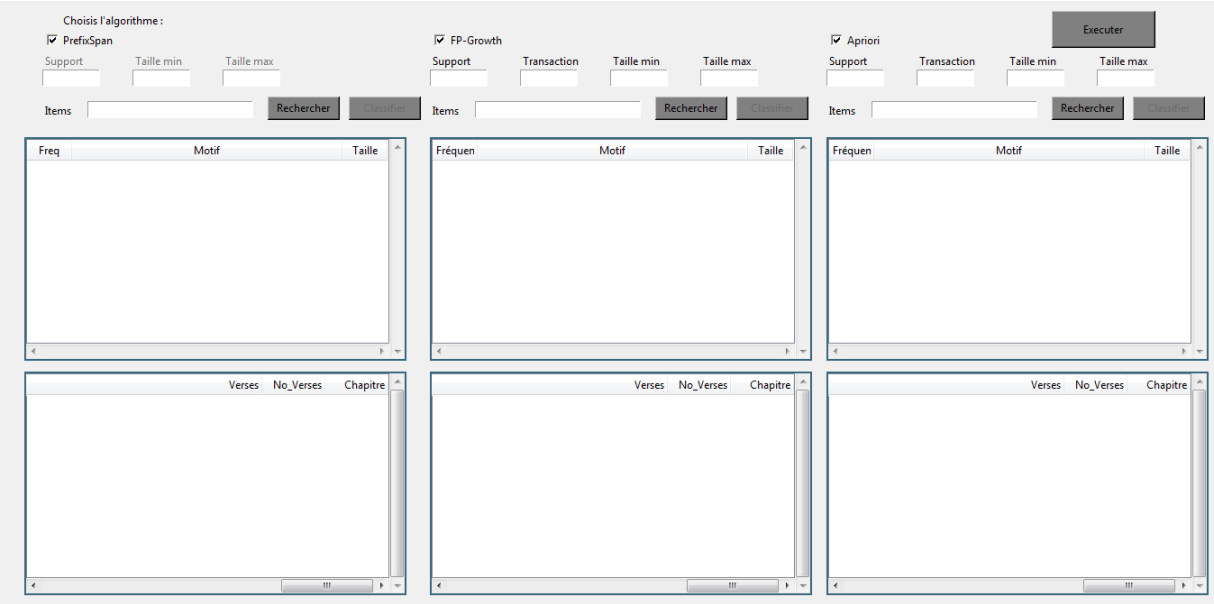


Figure 33: Interface de notre système

Interface de classification

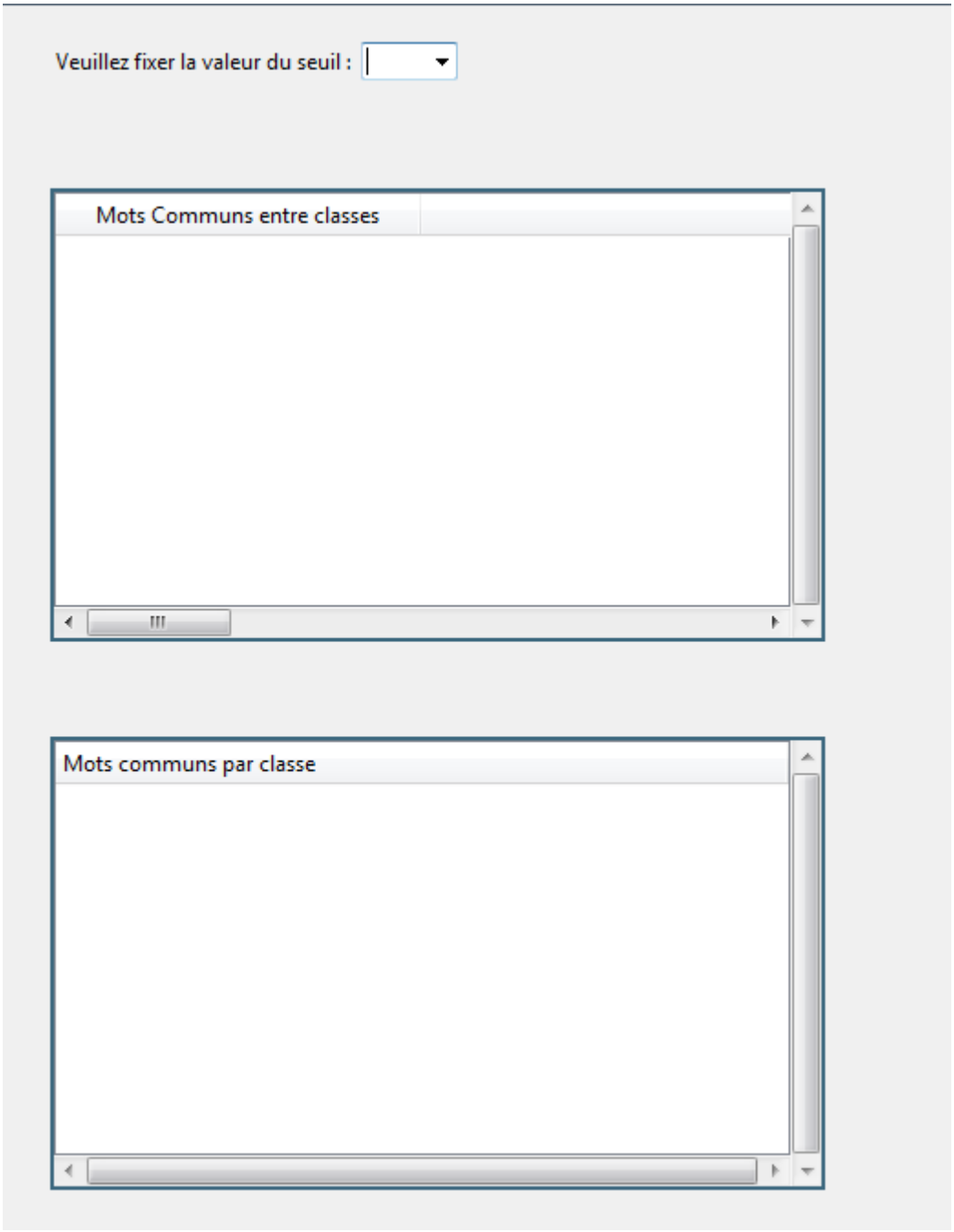


Figure 34: interface de classification

3.4 Conclusion

Dans ce chapitre nous avons présenté la démarche suivie dans notre projet avec ces différentes étapes et leurs processus de fonctionnement, ainsi l'environnement matériels et logiciels utilisé.

Chapitre 4

Analyse et discussion des résultats

4.1 Introduction

Dans ce chapitre nous allons discuter et faire un bilan sur les résultats obtenus.

4.2 Discussion des résultats

4.2.1 1ère expérimentation

Notre premier essai consiste à faire une recherche d'information avec l'algorithme Prefix-Span en choisissant un support=10, taille minimale=3, taille maximale=5 et un item vide.

Voici le résultat :

Freq	Motif	Taille
10	Ea`lim,gayob,$aha`dap	3
10	malakato,yamiyn,{ll~ah	3
10	*aAqu,Ea*aAb,kaAna	3
10	{fotaraY`,ka*ib,{ll~ah	3
10	nab~a>a,kaAna,Eamila	3
10	>aTaAEa,rasuwl,{ll~ah	3
10	basaTa,rizoq,$aA^'a	3
10	qa`tala,sabiyl,{ll~ah,{ll~ah	4
10	{botagaY`,{ll~ah,{ll~ah	3
10	A^xar,{ll~ah,{ll~ah	3

Figure 35: Résultat de la première expérimentation en buckwalter

Verses	No_Verses	Chapitre
لْخَبِيرُ ۝٧٣ وَلَهُ الْمُلْكُ يَوْمَ يُنفَخُ فِي الصُّورِ ۚ عَالِمُ الْغَيْبِ وَالشَّهَادَةِ ۚ وَهُوَ الْحَكِيمُ ٱوَهُوَ الَّذِي خَلَقَ السَّمَاوَاتِ وَالْأَرْضَ بِالْحَقِّ ۖ وَيَوْمَ يَقُولُ كُن فَيَكُونُ ۚ قَوْلُهُ الْح	73	6
○ الْغَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ ۝٩٤ أَنَا اللَّهُ ذُو أَخْبَارِكُمْ ۚ وَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ ثُمَّ تُرَدُّونَ إِلَىٰ عَالِمِ يَخْتَرُونَ إِلَيْكُمْ إِذَا رَجَعْتُمْ إِلَيْهِمْ ۚ قُل لَّا تَعْتَذِرُوا لَن نُّؤْمِنَ لَكُمْ قَدْ نَبَ	94	9
الْغَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ ۖ وَسَتُرَدُّونَ إِلَىٰ عَالِمِ	105	9
عَالِمُ الْغَيْبِ وَالشَّهَادَةِ الْكَبِيرُ الْمُتَعَالِ	9	13
عَالِمِ الْغَيْبِ وَالشَّهَادَةِ فَتَعَالَىٰ عَمَّا يُشْرِكُونَ	92	23
ذَٰلِكَ عَالِمُ الْغَيْبِ وَالشَّهَادَةِ الْعَزِيزُ الرَّحِيمُ	6	32
○بَادِكَ فِي مَا كَانُوا فِيهِ يَخْتَلِفُونَ قُلِ اللَّهُمَّ فَاطِرَ السَّمَاوَاتِ وَالْأَرْضِ عَالِمَ الْغَيْبِ وَالشَّهَادَةِ أَنتَ تَحْكُمُ بَيْنَ عِ	46	39
○يمُ هُوَ اللَّهُ الَّذِي لَا إِلَٰهَ إِلَّا هُوَ ۖ عَالِمُ الْغَيْبِ وَالشَّهَادَةِ ۖ هُوَ الرَّحْمَٰنُ الرَّح	22	59
غَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ قُلْ إِنَّ الْمَوْتَ الَّذِي تَفِرُّونَ مِنْهُ فَإِنَّهُ مُلَاقِيكُمْ ۖ ثُمَّ تُرَدُّونَ إِلَىٰ عَالِمِ الْ	8	62
عَالِمُ الْغَيْبِ وَالشَّهَادَةِ الْعَزِيزُ الْحَكِيمُ	18	64

Figure 36: un des résultats de préfixe-Span écrits en arabe

Nous remarquons que les motifs affichés sont d'une taille qui varie entre 3 et 5 mots, avec un support supérieur ou égal à 10.

4.2.2 2ème expérimentation

Dans cette expérimentation, nous avons fait la recherche d'information avec l'algorithme Prefix-Span aussi et avec les mêmes paramètres que la 1ère expérimentation, mais cette fois-ci avec un item : ' **rab~, kaAna**'

Voici le résultat :

Freq	Motif	Taille
13	jaA^'a,rab~,kaAna	3
10	daEaA,rab~,kaAna	3
54	qaAla,rab~,kaAna	3
10	rab~,kaAna,mu&omin	3
11	rab~,kaAna,kaAna	3
11	rab~,kaAna,{ll~ah	3
14	rab~,{ll~ah,kaAna	3
12	rab~,qaAla,kaAna	3
16	{ll~ah,rab~,kaAna	3

Figure 38: Résultat de la deuxième expérimentation en buckwalter

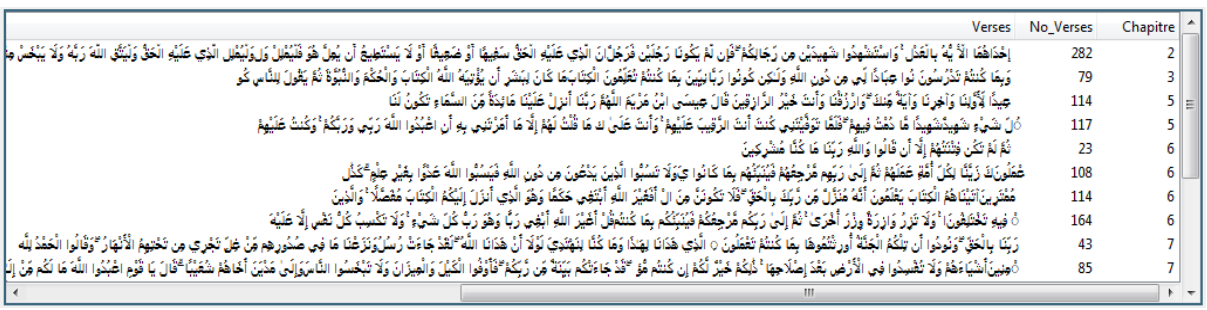


Figure 37: un des résultats de prefix-Span avec motif écrits en arabe

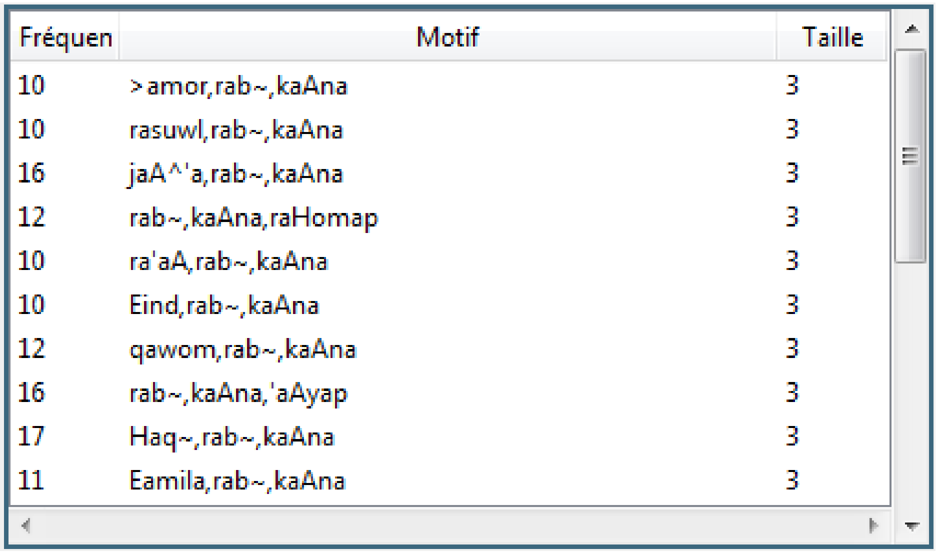
Nous remarquons que par rapport à la première expérimentation les résultats ont diminué, à cause de la recherche par mot.

4.2.3 3ème expérimentation

L'idée derrière cette expérimentation est de voir à quel point l'extraction d'information avec un algorithme qui se base sur la recherche des motifs non-séquentiels peut causer une différence des résultats par rapport à l'algorithme Prefix-Span qui est séquentiel.

Nous avons choisi l'algorithme Fp-Growth avec les paramètres : support=10, taille minimale=3, taille maximale=5 et item : : ‘ **rab~, kaAna**’

Voici le résultat :



Fréquen	Motif	Taille
10	>amor,rab~,kaAna	3
10	rasuwl,rab~,kaAna	3
16	jaA^'a,rab~,kaAna	3
12	rab~,kaAna,raHomap	3
10	ra'aA,rab~,kaAna	3
10	Eind,rab~,kaAna	3
12	qawom,rab~,kaAna	3
16	rab~,kaAna,'aAyap	3
17	Haq~,rab~,kaAna	3
11	Eamila,rab~,kaAna	3

Figure 39: Résultat de la troisième expérimentation

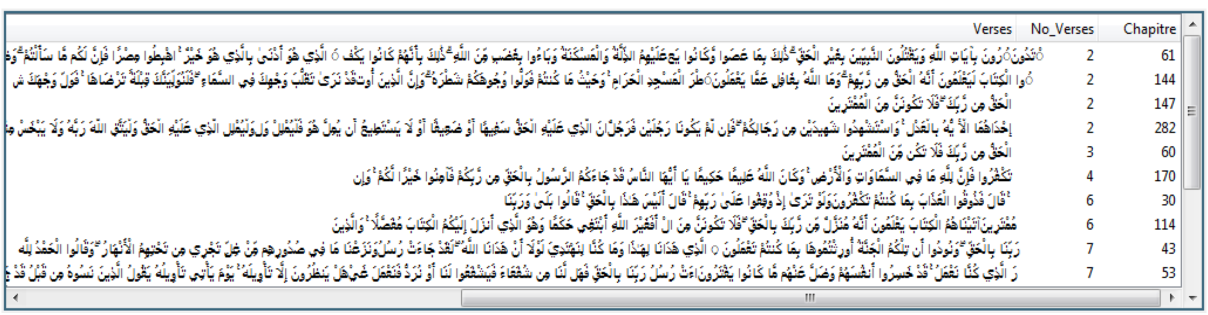


Figure 40: un des résultats de Fp-Growth avec motif écrits en arabe

De cette expérimentation nous remarquons que l'extraction des motifs séquentiels et non séquentiel, n'ont pas les mêmes résultats, d'où nous pouvons constater que l'ordre des motifs est un indicateur de comparaison entre les algorithmes.

4.2.4 Résultats de la classification

4.2.4.1 1ère expérimentation sans motif

Nous avons fait un test avec l'algorithme Prefix-Span et les paramètres suivants :

Support=10, Taille min=2, Taille max=5 et motif= rab~

Voici le résultat :

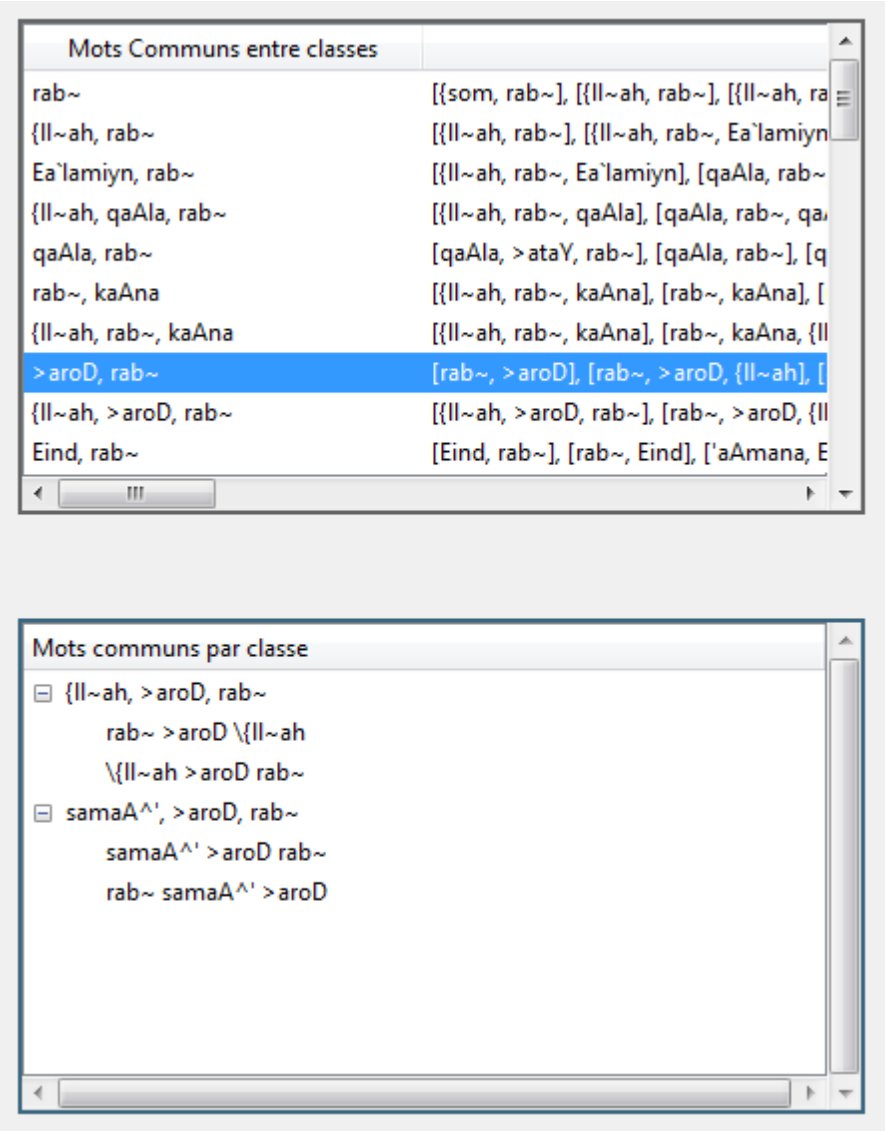


Figure 41: Résultat de la classification avec motif

Par la suite, nous avons réussi à ajouter une fonction qui permet de dessiner le graphe hiérarchique des classes similaires ayant le même mot(s) commun(s).

Voici un exemple :

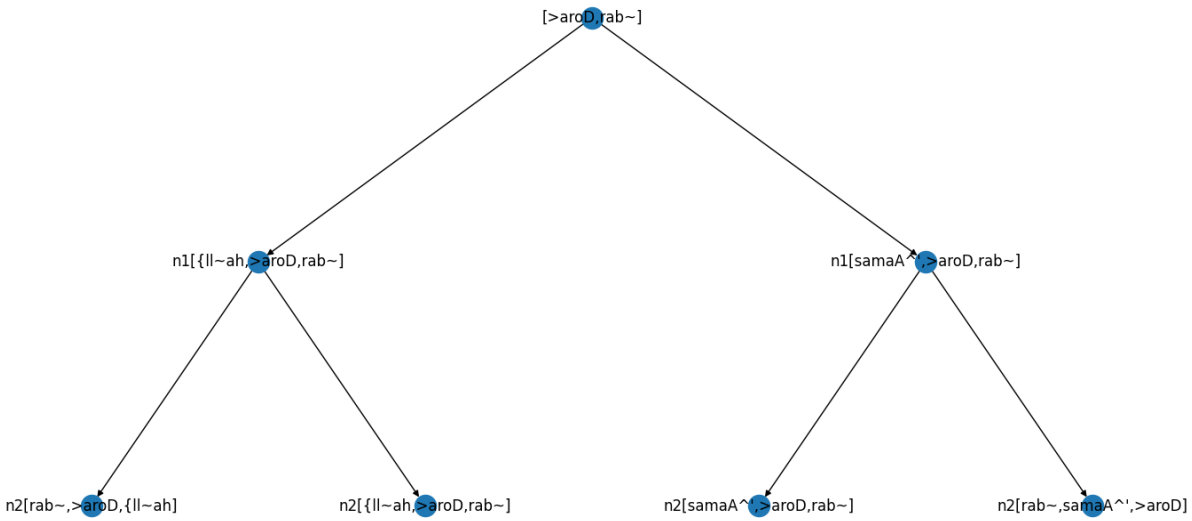


Figure 42: Graphe hiérarchique des classes qui ont le mot commun "aroD, rab~"

4.2.4.2 2ème expérimentation avec motif

Nous avons fait un test avec l'algorithme Prefix-Span et les paramètres suivants :

Support=30, Taille min=1, Taille max=3 et sans motif

Voici le résultat :

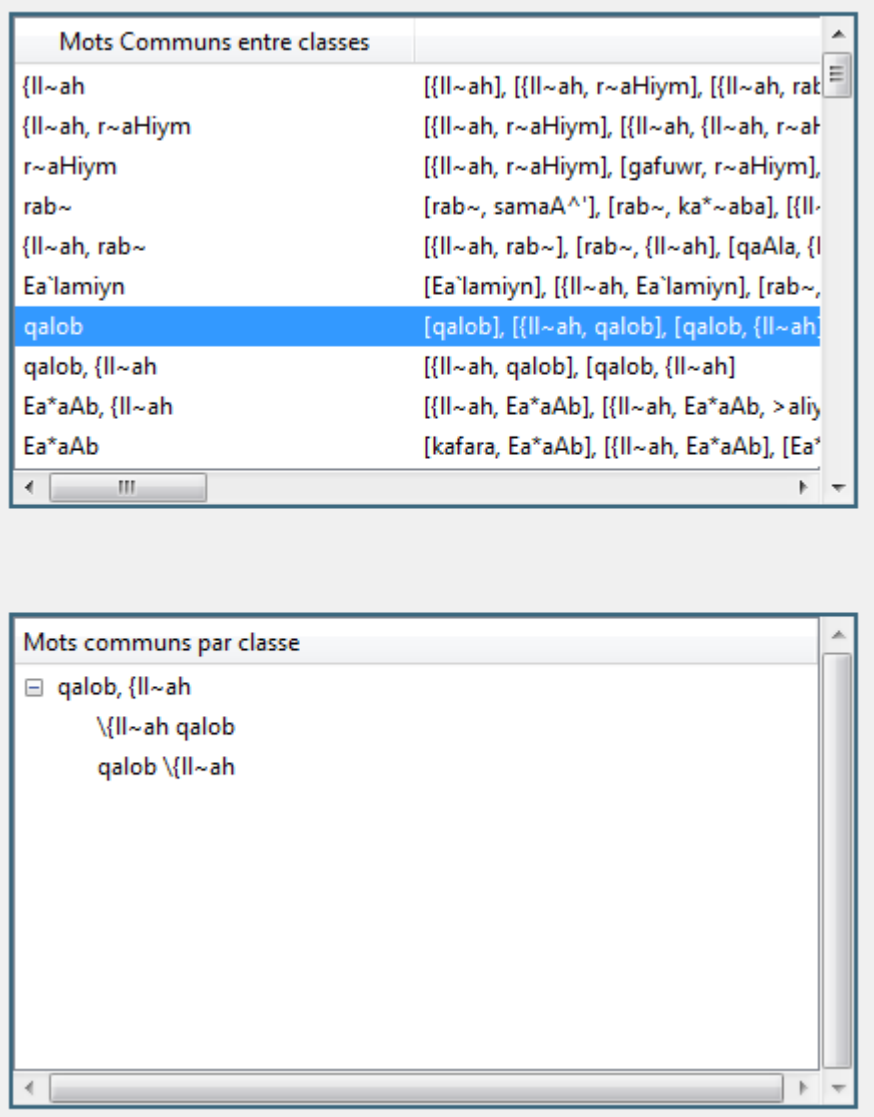


Figure 43: Résultat de la classification sans motif

Voici un graphe d'une des classes :

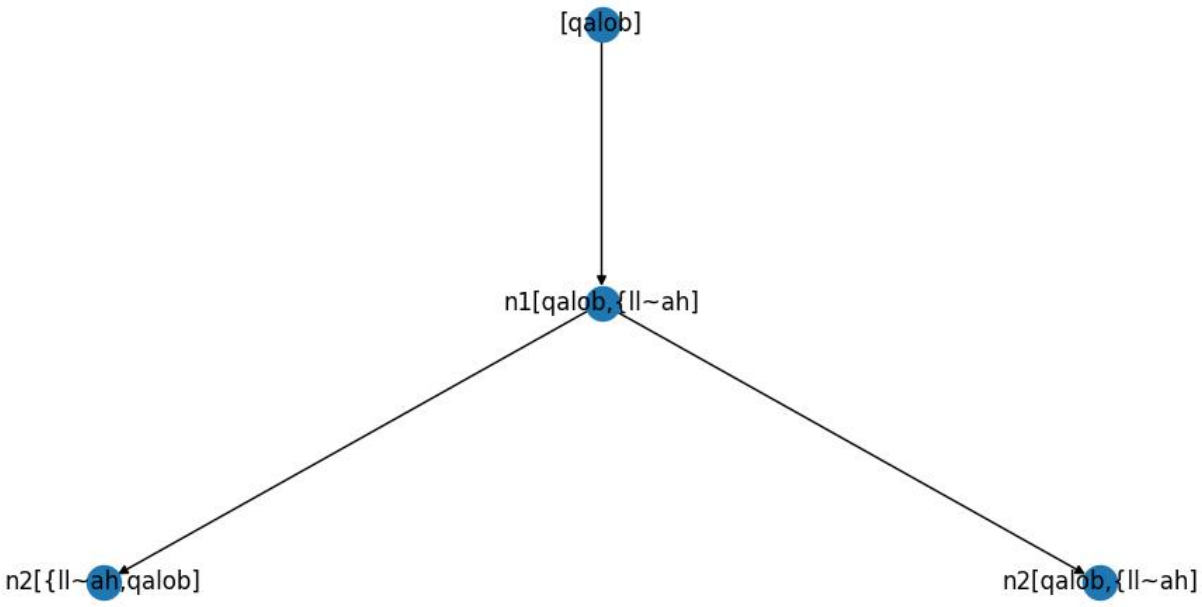


Figure 44: Graphe hiérarchique des classes ayant "qualob" comme mot commun

Après avoir fait plusieurs tests avec de différents seuils, allant de 0.1 vers 1, nous avons pu constater que en augmentant le seuil, les résultats diminuent.

Nous avons remarqué aussi que la taille du support influe la quantité de résultats. D'où y a une relation inverse entre la taille du support et le nombre de résultats. Plus le support est petit, plus le nombre des résultats est grand et le vice versa.

4.3 Conclusion

Dans ce chapitre nous avons analysé les résultats obtenus en faisant quelques expérimentations.

Conclusion Générale

Ce travail de mémoire porte sur la problématique d'extraction de connaissances à partir de textes en arabe, plus communément appelée la fouille de textes. Il s'articule autour des problèmes liés à l'analyse des textes, la fouille de textes proprement dite, et l'interprétation des éléments de connaissances extraits. Dans ce cadre, un système d'extraction des connaissances nécessaires pour analyser le coran en fonction de leur fréquence est étudié et implanté. Les méthodes de fouille de données appliquées sont la recherche de motifs fréquents séquentiels (avec l'algorithme « Prefix-Span »), de motifs fréquents non séquentiels (avec l'algorithmes « Fp-Growth & Apriori »).

Le mémoire s'attache à définir précisément le processus de fouille de textes et ses principales caractéristiques et propriétés en s'appuyant sur l'extraction de motifs fréquents et de règles d'association. En outre, Les objectifs de ce mémoire étaient multiples, la réalisation de ces objectifs nous a conduits à entamer plusieurs disciplines en même temps et à résoudre beaucoup de problèmes. La langue traitée dans le cadre de ce mémoire est lexiquement riche, morphologiquement complexe et d'un degré d'ambiguïté élevé

Ce travail a permis d'acquérir nos connaissances dans le domaine de la programmation Python, ainsi de conforter nos connaissances dans le domaine de science de données et tout ce qui concerne la fouille de texte.

Le mémoire inclut la réalisation d'un système appelé AYAT : « ايات » en arabe, ainsi quelques expérimentation ont été faites, pour faire l'analyse et discussion des résultats obtenus. D'où notre projet a pour but de faciliter la recherche et l'extraction d'information aux analystes et experts en domaine islamique.

Bibliographie

[1] Bastien L, https://www.lebigdata.fr/traitement-naturel-du-langage-nlp-definition
[2] Le site web : https://monkeylearn.com/text-mining/

[3]Le site web : https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501887-natural-language-processing-nlp-definition-techniques-et-modeles
[4] Matallah Hocine, Classification automatique de textes approche orientée agent ; UNIVERSITE université Aboubekr Belkaid-Tlemcen faculté des sciences département d'informatique.
[5] Article dans le site web : https://datascientest.com/text-mining-definition
[6] Pierre Holat, Fouille de motifs et modélisation statistique pour l'extraction de connaissances textuelles. Modélisation et simulation. Université Sorbonne Paris 2018.

[7] Marc Plantevit. Extraction De Motifs Séquentiels Dans Des Données Multidimensionelles. Informatique [cs]. Université Montpellier II - Sciences et Techniques du Languedoc, 2008. Français. fftel00319242

[8] Jian Pei, Member, IEEE Computer Society, Jiawei Han, Senior Member, IEEE,Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, Member, IEEE Computer Society, and Mei-Chun Hsu Mining sequential Patterns by Pattern-Growth:The PrefixSpan Approach

[9] Article dans le site web :https://www.softwaretestinghelp.com/apriori-algorithm/

[10] Article dans le site web :https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining

[11] A.Kaheel, «Chercheurs dans les miracles du saint coran,» [En ligne]. Available: http://www.kaheel7.com/fr/

[12] Dataset utilisé **Quranic Arabic Corpus** : https://corpus.quran.com/

[13] K. Khoukha, «Extraction intelligente des motifs à partir du texte en arabe,» 2020/2021

[14] Elsa Negre, Comparaison de textes : quelques approches.... 2013. ffhal-00874280

Download Python3 https://www.python.org/downloads/
Download Visual Studio Code : https://code.visualstudio.com/
Download Kaheel Application : http://kaheel7.org/KaheelFiles/apps/13-10-2021.zip