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

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

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

# Système d'Indexation et d'Exploration Sémantique-Coranique dans les Textes Arabes

**Etudiante :**
**SEFIANE Fatiha**

**Encadrant:**
**Dr. BRAHMI Abderrezak**

**Année Universitaire 2015/2016**

# Dédicaces

*On dit souvent que la vie est une rose …. Et que le travail en est le miel …….*

*Et tous ce qui travaille …….auront peut-être l'estime*

*Et le respect d'autrui…….*

*Que ce modeste travail soit ma façon d'exprimer ma gratitude et mon dévouement ! A tout ce qui contribué à ma réussite….ainsi je dédie le fruit de mes efforts a :*

*Ma mère,*

*Mon père,*

*Mes frères Houcine et Mohamed,*

*Mes sœurs Fatima et Djawhara,*

*Toute ma famille*

*Mes amies Kawther,Nadjet,Lilia,Imene,Asma,Ghizlen,Ahlem,Meriem,Mansoria et Malika.*

*Tous mes enseignants et mes collègues à l'université de Mostaganem*

*Tous ceux que j'aime*

*Tous ceux qui m'aiment*

*Qu'ils trouvent ici l'expression de toute ma reconnaissance.*

**Fatiha.**

# REMERCIEMENTS

*Tout d'abord, Je tiens à remercier ALLAH le tout puissant et le miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.*

*Je tien à adresser également mes remerciements à mon encadreur* ***Dr. BRAHMI Abderrezak*** *qui a bien voulu mettre leur incomparable savoir et expériences à ma disposition.*

*Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.*

*Je garde une place toute particulière à mes parents, mes frères et mes sœurs qui sont toujours à mes cotés.*

*Je tiens aussi à remercier du fond du cœur Mlle Berrezoug Asma, doctorante à l'université de Mostaganem ainsi que M. Boudchiche Mohamed,* doctorant au *laboratoire Informatique à l'Université Mohamed Premier-Oujda, Maroc.*

*Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.*

*Merci à tous et à toutes.*

# Résumé

La recherche d'information moderne sollicite de plus en plus des techniques d'intelligence artificielle et des modèles efficaces pour l'analyse et l'indexation dans les langues morphologiquement complexes. Par ailleurs le texte coranique représente une référence terminologique et ontologique incontestable pour la langue arabe.

L'objectif de ce projet est de réaliser un système de navigation intelligente dans les textes arabes en exploitant la richesse linguistique du Saint Coran dans une indexation sémantique. Notre contribution consiste en la réalisation d'une plate-forme d'indexation et de recherche dans le texte arabe. L'architecture proposée pour le stockage, l'indexation et la recherche des unités morphologique dans le Coran, nous a permis de concrétiser un modèle de référencement linguistique. Cette plate-forme représente une phase importante pour aboutir à l'indexation et le référencement sémantique des textes arabes.



**Mots-Clés**

Recherche d'information, Analyse de l'arabe, Exploration, Sémantique Coranique, Modèle de référencement.



# Abstract

Modern information retrieval needs more artificial intelligence techniques and effective models for analysis and indexing in morphologically complex languages. Moreover, the Qur'an is a terminological and ontological indisputable reference to the Arabic language.

The objective of this project is to achieve an intelligent navigation system in Arabic texts by exploiting the linguistic richness of the Holy Quran in a semantic indexing. Our contribution is the realization of a platform for indexing and searching in the Arabic text. The proposed architecture for storing, indexing and retrieval of morphological units in the Qur'an, has allowed us to realize a linguistic referencing model. This platform represents an important step to achieve semantic indexing and referencing of Arabic texts



**Key-words**

Information Retrieval, Arabic Analysis, Exploration, Quran Semantics, Referencing Model.



# ملخص

إن البحث الحديث عن المعلومات يتطلب المزيد من تقنيات الذكاء الاصطناعي الفعالة لتحليل وفهرسة اللغات ثرية الاشتقاق والصرف. وعلاوة على ذلك فإن القرآن الكريم يمثل مرجعا لغويا ودلاليا مضبوطا للغة العربية.

والهدف من هذا المشروع هو إنتاج نظام ذكي لاستكشاف النصوص العربية من خلال استغلال الثراء اللغوي للقرآن الكريم في الفهرسة الدلالية. مساهمتنا تتمثل في إنجاز أرضية للفهرسة والبحث في النص العربي. التصميم المقترح، للتخزين والفهرسة و البحث عن المفردات بأشكالها المتنوعة في القرآن، مكننا من تجسيد نموذج للإحالة اللغوية. هذه الأرضية تمثل خطوة هامة للوصول إلى الفهرسة والإحالة الدلالية للنصوص العربية.



**الكلمات مفتاحيه**

البحث عن المعلومات، تحليل اللغة العربية، الاستكشاف، الدلالات القرآنية، نموذج الإحالة.

# TABLES DES MATIÉRES

# LISTE DES FIGURES

# LISTE DES TABLEAUX

# LISTE DES ABRÉVIATIONS

***AWN*** **:** Arabic Word Net.

***DDL*** **:** Data definition language.

***DCL*** **:** Data control language.

***DML*** **:** Data manipulation language.

***DQL*** **:** Data query language.

***EDI*** **:** Integrated Development Environment.

***LDC*** **:** Linguistic Data Consortium.

***MCD*** **:** Model Conceptuel de Données.

***RB*** **:** Réseaux Bayésiens.

***RI*** **:** Recherche d'Information.

***RLSC*** **:** Le classifieur à moindre carrés régularisés.

***RNs*** **:** Les réseaux de neurones.

***RSV*** **:** Retrieval Status Value.

***SQL*** **:** Structured Query Language.

***SRI*** **:** Système de Recherche d'Information.

***SVM*** **:** les séparateurs à vaste marge.

***TI*** **:** technologie d'information.

***UML*** **:** Unified Modeling Language, « langage de modélisation unifié ».

***WN*** **:** Word Net.

# INTRODUCTION GÉNÉRALE

A travers toute son histoire, l'homme avait toujours besoin de l'information pour prendre les meilleures décisions possibles. Avec les nouvelles technologies, l'information numérique occupe de jour en jour le centre de nos activités. En particulier, l'accès à l'information est devenu un besoin vital dans notre vie quotidienne. Les systèmes de recherche d'information ﴾SRI﴿ sont conçus pour faciliter l'accès aux informations stockées et répondre convenablement à nos besoins en information. Il peut s'agir d'une image, d'une vidéo, d'une position géographique, …etc. Mais incontestablement, le texte reste l'objet de recherche le plus répandu. En effet, la recherche dans le fond documentaire s'avère d'une utilité et d'une diversité exceptionnelle. De part la précision de la connaissance contenue, le texte est très sensible à la langue et sa recherche nécessite, par conséquent, des prétraitements spécifiques.

Les techniques du traitement automatique des langues permettent d'extraire des textes des informations plus riches que de simples unités lexicales. Ces informations de nature morphologique, syntaxique et sémantique ont été partiellement utilisées en RI pour améliorer les méthodes d'appariement, les représentations des contenus des documents et requêtes et le processus de recherche.

L'arabe, une des six langues officielles des Nations Unies, est la langue maternelle de plus de 300 millions de personnes[1]. Le domaine recherche d'information arabe, devenu un centre de la recherche et du développement commercial, est du à la nécessité essentielle de tels outils pour des personnes dans l'ère électronique. Le nombre d'internautes arabophones est en pleine expansion et connait la plus haute croissance durant la dernière décennie. Cependant, peu de moteurs de recherche sont mis à la disposition des utilisateurs arabophones et même, les moteurs standards existants ou intégrés dans certaines applications, gèrent mal la variation typographique du texte arabe.

En effet, la langue arabe s'appui sur une morphologie fortement flexionnelle, dérivationnelle et agglutinante ; elle se caractérise par l'absence des voyelles courtes ﴾diacritiques﴿ dans la plupart des textes écrits. Contrairement à l'anglais ou le français, les voyelles courtes arabes ne sont pas des lettres de l'alphabet, ce sont des signes diacritiques qui se rajoutent aux consonnes ﴾lettres﴿ et qui jouent le même rôle que les voyelles dans les autres langues. Généralement les écrits arabes ne sont pas vocalisés et c'est au lecteur de deviner les diacritiques des textes au moment de la lecture. En revanche, le texte religieux ﴾texte coranique﴿ est entièrement vocalisé, néanmoins l'habitude de rédiger sans diacritiques s'applique aussi à l'utilisateur lorsqu'il exprime sans besoin en information depuis le texte coranique.

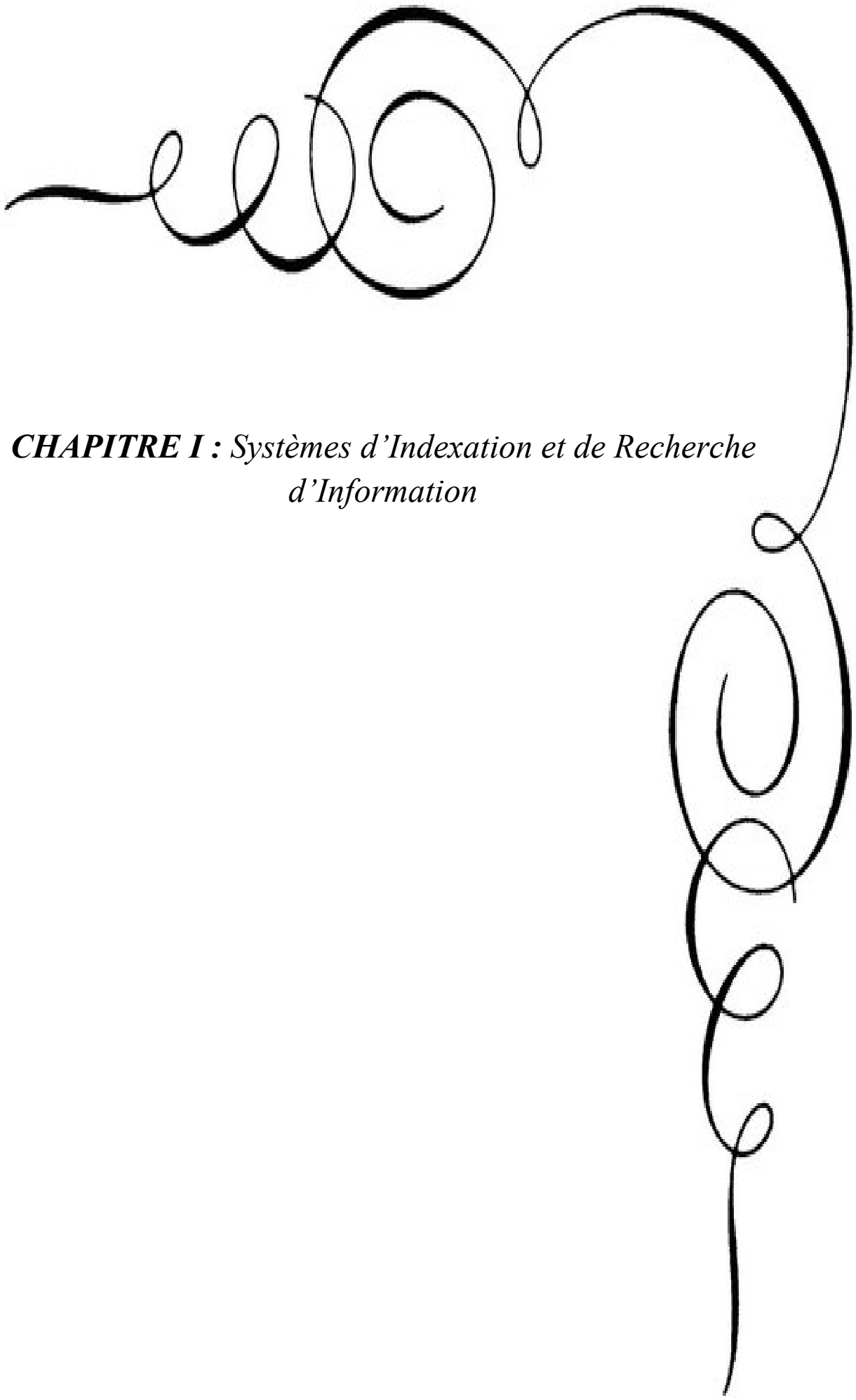
[1] http://www.frcu.eun.eg/www/homepage/cdc/cdc.htm. Accédé le 12/03/2016.

Depuis l'apparition de la première copie numérique du coran, il y a eu un effort considérable pour produire un texte précis du Coran, mais en raison de certaines difficultés comme le manque des signes diacritiques, ces efforts ont échoué dans la plupart des cas, et malheureusement, les textes du coran sont apparus dans la majorité des sites web et des applications coraniques souffraient de beaucoup d'erreurs et fautes de frappes. Cette situation terrible permettait de lancer le projet Tanzil, de produire un texte de Coran sans erreur vérifié minutieusement, et faire de ce texte disponible sur les sites Web du Coran et des applications pour empêcher la propagation ultérieure des textes erronées du Coran.

Aussi naturel qu'indispensable, les outils de recherche d'information devraient accompagner toute application dans le fond documentaire du Coran. Des tâches telles que le filtrage ou la recherche ad-hoc par requête libre s'avèrent primordiale dans le contexte coranique. D'autres projets très ambitieux, liés aux tâches intelligentes de l'extraction de connaissances, nécessitent des techniques avancées du Text-mining mais bien plus, des outils adéquats de prétraitement linguistique.

Le Coran encapsule le cœur de la langue arabe et par conséquent, il doit nous fournir l'ontologie linguistique et sémantique idéale. Cette croyance est loin d'être validée par les modèles d'intelligence artificielle. Dans cette perspective, l'idée du référencement coranique propose une approche simple et pragmatique pour exploiter le texte coranique. Un expert ou un simple utilisateur souhaite pour n'importe quel texte, lu sur un support électronique, voir, partiellement ou totalement, les versets qui lui sont associés. Ce référencement s'appui principalement sur la terminologie déployée dans le texte arabe. Une fois réalisé, l'approche peut être développée vers un référencement sémantique basé sur les techniques du text-mining.

Le présent mémoire décrit la démarche de conception et de réalisation d'un système d'indexation et d'exploration sémantique-coranique dans les textes arabes. Le manuscrit est organisé en quatre chapitres **:** Nous abordons dans le premier chapitre la recherche d'information et décrivant les différents modèles d'indexation et les méthodes de calcul de correspondance. Le second chapitre, expose les caractéristiques morphologiques de la langue arabe et les problèmes liés à son traitement automatique. Nous y réservons une partie pour la description du texte Coranique et les aspects techniques nécessaires pour ses supports numériques. Nous enchainons avec un troisième chapitre qui porte sur l'architecture globale et les aspects conceptuels de notre système. Le dernier chapitre présente les techniques d'implémentation et certains résultats obtenu, avant de clôturer ce manuscrit par une synthèse de l'essentiel de notre travail et tracer les perspectives pour le développer.



# ***CHAPITRE I :*** *Systèmes d'Indexation et de Recherche d'Information*

## I.1. Introduction

La ‹RI› peut être définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations. Le défi est de pouvoir, parmi le volume important de documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur.

L'opération de la ‹RI› est réalisée par des outils informatiques appelés ‹SRI›, ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur ‹requête› avec une représentation du contenu des documents ‹fiche ou enregistrement› au moyen d'une fonction de comparaison ‹ou de correspondance›. L'essor du web a remis la ‹RI› face à de nouveaux défis d'accès à l'information, il s'agit cette fois de retrouver une information pertinente dans un espace diversifié et de taille considérable. Ces difficultés ont donné naissance à une nouvelle discipline appelée Recherche d'Information sur le Web.

## I.2. Recherche d'information

### I.2.1. Définition

**Déf1 :** Salton définit la ‹RI› comme la branche de l'informatique qui consiste à acquérir, organiser, stocker, rechercher et sélectionner l'information **[1]** . Les domaines d'application de la ‹RI› sont:

- Internet
- Bibliothèques numériques «digital library»
- Entreprises

**Déf2 :** La ‹RI› est un domaine qui s'intéresse à la représentation, le stockage, l'organisation et l'accès aux éléments d'information. Elle étudie la manière de répondre pertinemment à une requête pour retrouver de l'information utile dans une collection de données. Cette dernière est constituée de documents, d'une ou de plusieurs bases de données, décrits par un contenu ou des métadonnées associées. Le contenu des documents peut être un texte, une page Web, une image, un son, une vidéo ou même un élément spatial d'une carte géographique. On parle ainsi de plusieurs sous-disciplines plus fines telles que la recherche du texte, la recherche sur le Web, la recherche d'image, la recherche multimédia ou la recherche d'information géographique. Notre étude s'intéresse seulement à la recherche dans le contenu textuel **[2]**.

### I.2.2. Concepts de base de la recherche d'information

La ‹RI› est considérée comme l'ensemble des techniques permettant de sélectionner à partir d'une collection de documents, ceux qui sont susceptibles de répondre aux besoins de l'utilisateur. La gestion de ces informations implique le stockage, la recherche et l'exploration des documents pertinents. De ce contexte plusieurs concepts clés peuvent être définis, nous avons donc trouvé utile de les clarifier.

### I.2.2.1. Collection de document

La collection de documents «ou fond documentaire» constitue l'ensemble des informations exploitables et accessibles. Elle est constituée d'un ensemble de documents. Dans le cas général et pour un souci d'optimalité, la base constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telles sortes que la gestion «ajout, suppression d'un document» ou l'interrogation «recherche» de la base se font dans les meilleures conditions de coût.

### I.2.2.2. Document

Le document constitue l'information élémentaire d'une collection de documents. L'information élémentaire, appelée aussi granule de document, peut représenter tout ou une partie d'un document.

### I.2.2.3. Besoin d'information

La notion de besoin en information en recherche d'informations est souvent assimilée au besoin de l'utilisateur. Trois types de besoin utilisateur ont été définis par **[3]:**

**Besoin vérificatif :** l'utilisateur cherche à vérifier le texte avec les données connues qu'il possède déjà. Il recherche donc une donnée particulière, et sait même souvent comment y accéder. La recherche d'un article sur Internet à partir d'une adresse connue serait un exemple d'un tel besoin. Un autre exemple serait de chercher la date de publication d'un ouvrage dont la référence est connue. Un besoin de type vérificatif est dit stable, c'est-à-dire qu'il ne change pas au cours de la recherche.

**Besoin thématique connu :** l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connus. Un besoin de ce type peut être stable ou variable : il est très possible en effet que le besoin de l'utilisateur s'affine au cours de la recherche. Le besoin peut aussi s'exprimer de façon incomplète, c'est-à-dire que l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête mais seulement un sous-ensemble. C'est ce qu'on appelle dans la littérature le label.

**Besoin thématique inconnu :** cette fois, l'utilisateur cherche de nouveaux concepts ou de nouvelles relations en dehors des sujets ou des domaines qui lui sont familiers. Le besoin est intrinsèquement variable et est toujours exprimé de façon incomplète.

### I.2.2.4. Requête

La requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le «SRI» et l'utilisateur. Divers types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique.

### I.2.2.5. Modèle de représentation

Un modèle de représentation est un processus permettant d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique. Ce processus de conversion est appelé indexation. Le résultat de l'indexation constitue le descripteur du document ou de la requête, qui est une liste de termes ou groupes de termes ‹concepts›, significatifs pour l'unité textuelle correspondante, auxquels sont associés généralement des poids, pour différencier leurs degrés de représentativité du contenu sémantique de l'unité en question. L'ensemble des termes reconnus par le ‹SRI› est rangé dans une structure appelée dictionnaire constituant le langage d'indexation. Ce type de langage garantit le rappel de documents lorsque la requête utilise dans une large mesure les termes du dictionnaire. En revanche, il y a risque important de perte d'informations lorsque la requête s'éloigne de ce vocabulaire.

### I.2.2.6. Modèle de recherche

Il représente le modèle du noyau d'un ‹SRI›. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer. Il est utilisé pour la recherche d'informations proprement dite et est étroitement lié au modèle de représentation des documents et des requêtes.

### I.2.3. Les applications relatives à la RI

La ‹RI› est un domaine vaste qui se situe dans les frontières de plusieurs disciplines tel que :

### I.2.3.1. Recherche adhoc

La recherche est dite **ad-hoc,** lorsque la requête est introduite par l'utilisateur librement dans un langage naturel.

### I.2.3.2. Classification /catégorisation ‹clustering›, Question-réponses ‹Query answering›

La tâche de catégorisation des textes consiste à classer automatiquement un ensemble de documents selon des catégories prédéfinies. Cette approche est attractive du moment où elle décharge les organisations des tâches fastidieuses de classification manuelle des documents **[4]**. Elle représente une alternative efficace, à titre de pré-organisation, des catalogues de recherche dans les bibliothèques électroniques.

La classification automatique des documents fait intervenir les méthodes d'apprentissage automatique avec les modèles de la ‹RI› en vue d'offrir à l'utilisateur une navigation thématique guidée dans les bibliothèques électroniques spécialisées. Les méthodes d'apprentissage supervisé ‹binaire, multi-classe ou multi-étiquette› trouvent dans les applications de catégorisation de texte l'un des premiers challenges pour évaluer leur performance. La classification Bayésienne naïve,

les réseaux de neurones ou les machines à vecteurs de support ﴾SVM﴿ constituent des techniques épandues pour cette tâche.

Pour résoudre le problème de classification supervisée, plusieurs approches ont été proposées et appliqués parmi lesquels nous citons:

Les méthodes à base de modèle statistique tel que les ﴾RB﴿ et le modèle de Markov ﴾HMMs﴿,

Les méthodes d'apprentissage par approche connexionniste telles que les ﴾RNs﴿,

Les méthodes d'apprentissage à base de noyau telles que ﴾SVM﴿ et ﴾RLSC﴿.

### I.2.3.3. Filtrage d'information ﴾filtering/recommendation﴿

Le filtrage d'information vise à extraire au sein d'un important volume d'informations générées dynamiquement, les documents susceptibles de correspondre aux intérêts de l'utilisateur. L'élimination du courrier indésirable, le blocage des expéditeurs malveillants ou le filtrage des sites sensibles représentent des domaines d'application très sollicitées. Le filtrage intègre aussi les opérations d'exploitation et de présentation des résultats. Les sources de filtrage sont dynamiques et évoluent dans le temps **[2]**.

Contrarient à une recherche ad-hoc, l'outil de filtrage permet de repérer exclusivement les documents relatifs aux centres d'intérêt ﴾requêtes préétablies﴿ formulés par l'utilisateur sous forme de sélection thématique prédéfinie**.**

Deux types de filtrage peuvent être cités dans ce contexte : le filtrage par contenu et le filtrage collaboratif. Ce dernier est devenu un sujet de recherche d'actualité pour répondre aux besoins grandissants dans le commerce électronique et les réseaux sociaux. Il repose sur les opinions d'un groupe d'utilisateurs pour recommander les objets ﴾produits, amis, sites …etc.﴿ les appropriés. Un aperçu détaillé des techniques utilisées pour cette approche peut être consulté dans **[5].**

### I.2.3.4. Résumé automatique ﴾Summarization﴿

Un résumé est une transformation réductive du texte source par sélection et/ou généralisation de ce qui est le plus significatif du texte original. Trois étapes essentielles doivent être considérées dans cette approche : l'identification des thèmes, l'interprétation et la génération **[6].**

La détection des phrases pertinentes dans le texte constitue la fonction clé dans ce genre de système qui épuise des techniques du traitement automatique du langage naturel et offre une alternative efficace pour la consultation des quantités immenses d'information en-ligne.

Le résumé automatique peut être réalisé à partir d'un seul document comme il peut être multi- document. Bien que l'appréciation des résumés générés soit assez sensible et compliquée, la méthode d'évaluation ROUGE est adoptée par la majorité de chercheurs **[2]**.

### I.2.3.5. Autres applications

- Méta-moteurs ﴾data-fusion, Meta-search﴿ ;

- Croisement de langues ‹cross language› ;
- Fouille de textes ‹Text mining›.

## I.3. La recherche web

La recherche sur le Web représente, de nos jours, l'application la plus importante pour la ‹RI›. Du point de vue de l'utilisateur, un système de recherche doit réaliser trois tâches :

- L'acquisition et l'analyse de la requête,
- Filtrage des documents pertinents,
- Visualisation des résultats ‹**voir Figure 1**›.

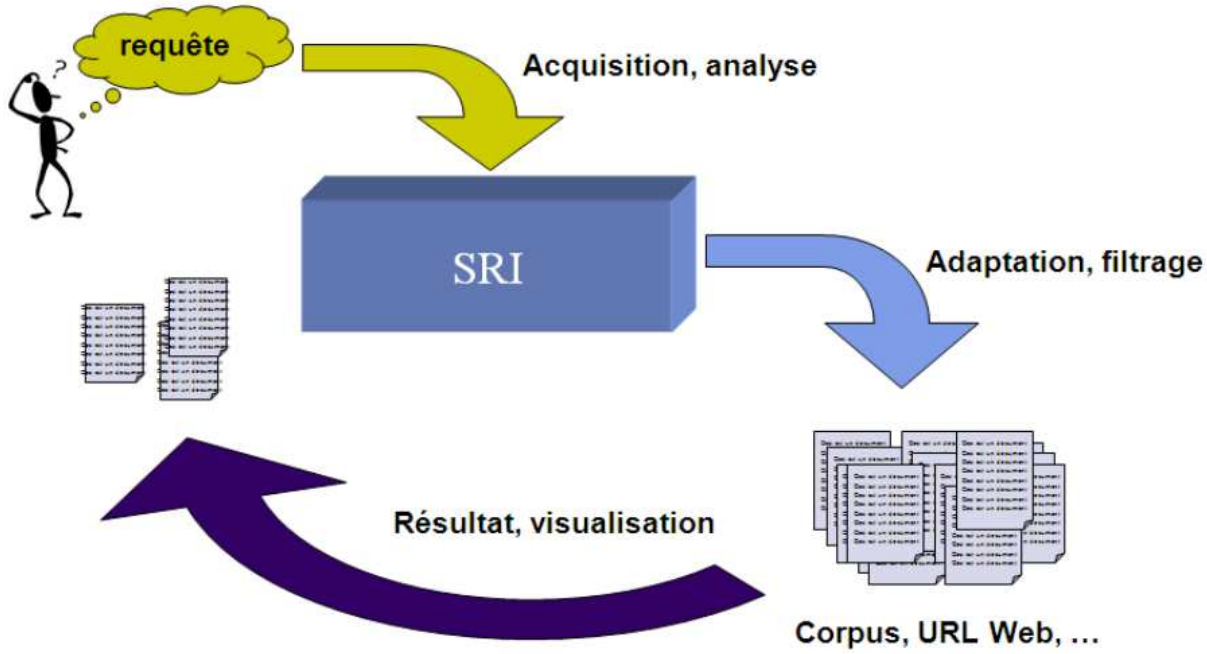



**Figure 1.** Système de recherche d'information ‹vue de l'utilisateur›**.**

### I.3.1. Définition

Un ‹SRI› est un système informatique qui permet de retourner à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin en informations d'un utilisateur, exprimé à l'aide d'une requête **[2]**.

### I.3.2. L'architecture d'un SRI

Afin de satisfaire les besoins grandissants et spécifiques des utilisateurs dans un environnement aussi riche que le Web, un moteur de recherche doit considérer d'autres fonctionnalités aussi indispensables que complexes. Nous considérons dans le présent contexte, l'architecture plus généralisées d'un moteur de recherche Web mais qui comprend les fonctionnalités de base d'un système classique. La collecte, la représentation et le calcul

des correspondances constituent le cœur de tout système de recherche en-ligne. On peut résumer ces fonctions comme suit :

- La collecte des documents et pages Web accessibles ‹crawling›,
- L'analyse du contenu documentaire
- L'indexation et la représentation,
- L'analyse et la représentation de la requête utilisateur,
- L'appariement et l'ordonnancement ‹ranking›,
- La visualisation.

L'architecture appropriée à ce genre de système est en "U" ‹**Figure 2**›. Ceci implique que les techniques d'analyse linguistique, ainsi que les modèles de représentation, doivent être équivalents aussi bien pour les documents que pour les requêtes.

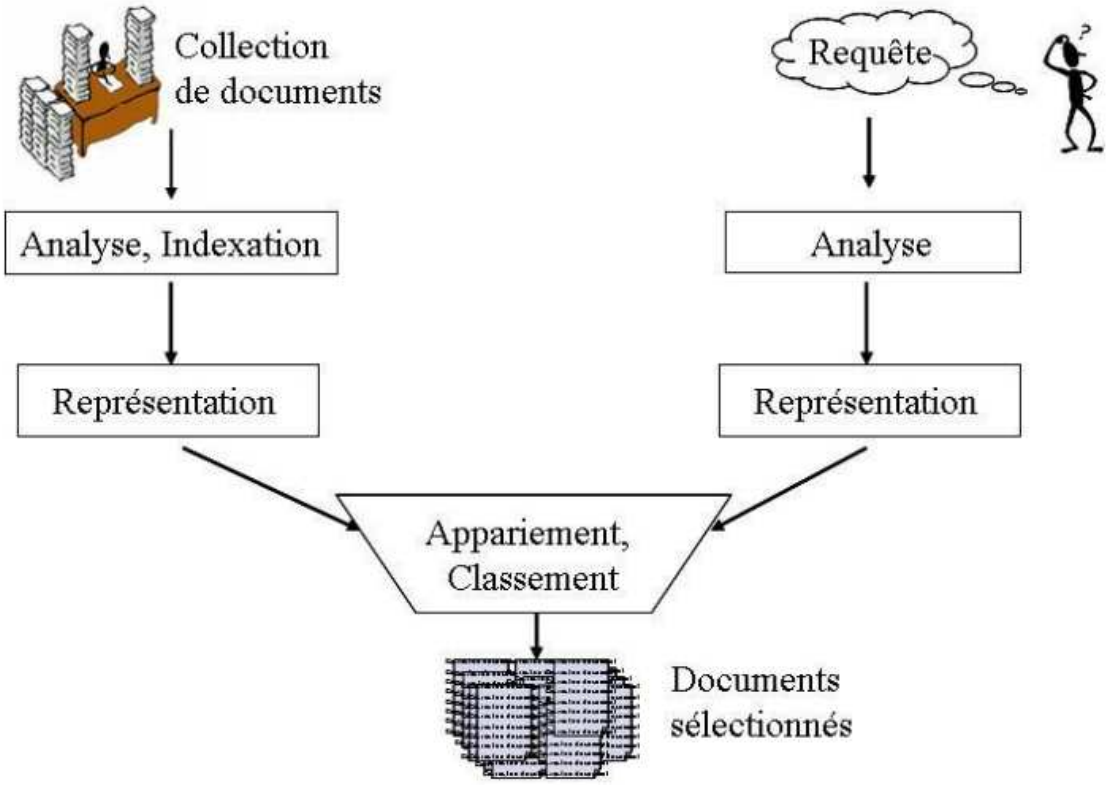



**Figure 2.** Architecture globale d'un système de recherche d'information.

### I.3.3. Fonctionnement

Le processus de recherche dans un système de recherche en ligne peut être décrit par les fonctions suivantes :

#### I.3.3.1. La collecte des documents

Elle est réalisée par exploration continue du Web ‹Web-crawler›. Un robot ‹agent parcourant de façon régulière les adresses d'Internet› est chargé de répertorier les nouveaux contenus mis el ligne. En suivant récursivement les liens hypertextes, un crawler construit graduellement l'index brut du moteur de recherche **[2].**

### I.3.3.2. L'analyse du contenu

L'analyse du contenu documentaire déploie des techniques de prétraitement linguistique pour sélectionner les contenus significatifs lexicalement. La détection préalable du codage des caractères et de la langue utilisée décide de la manière à suivre pour cette fonction. Les méthodes d'analyse automatique du langage naturel sont très sollicitées et la performance des systèmes de recherche en dépende fortement **[2].**

### I.3.3.3. La représentation

La représentation du texte respecte un modèle de document qui affectera directement la performance et l'efficacité de tout système de recherche. La fonction d'indexation a pour rôle d'extraire d'un document ou d'une requête, une représentation paramétrée ‹descripteurs› qui couvre au mieux son contenu significatif. Les descripteurs représentent généralement les termes reconnus par le système et sont rangés dans un dictionnaire constituant le langage d'indexation **[2].**

### I.3.3.4. L'appariement

L'appariement est basé sur une fonction d'inférence qui permet d'associer à une requête les documents pertinents. Etroitement liée au modèle de représentation, l'objectif de cette tâche est de sélectionner parmi des millions de documents seulement quelques dizaines ou centaines les plus appropriés.

La valeur de pertinence du système est définie comme mesure de similarité entre une requête ‹Q› et un document ‹d› du corpus et dénotée généralement par RSV‹d,Q›.La fonction de calcul de la RSV dépend du modèle de document utilisé notamment de la façon avec laquelle les termes sont pondérés. Elle permet dans la majorité des modèles, non seulement de sélectionner les documents pertinents à une requête, mais aussi de les ordonner par degrés de correspondance. L'astuce d'attribuer un score de pertinence, pour classer l'ensemble des documents par ordre décroissant de pertinence à une requête, a fait sa meilleure concrétisation avec le PageRank de Google **[2].**

### I.3.3.5. La reformulation de la requête

Cette étape peut paraître facultative mais elle peut améliorer considérablement la qualité d'un ‹SRI›. Celui-ci, qui se base essentiellement sur la requête exprimée par l'utilisateur, doit répondre efficacement au besoin d'information. Néanmoins, ce besoin n'est pas toujours clairement et explicitement formulé. Par conséquent, les documents retournés par le ‹SRI› peuvent appartenir à des domaines tout à fait divergents du centre d'intérêt de l'utilisateur.

La reformulation des requêtes consiste généralement à enrichir la requête de l'utilisateur en ajoutant des termes permettant de mieux exprimer son besoin **[7].** Néanmoins l'amélioration des résultats de recherche dépend du corpus lui-même et du nombre et de la façon avec laquelle les termes sont rajoutés.

### I.3.3.6. La visualisation

La forme la plus simple, pour afficher les résultats d'une recherche Web, consiste à inclure une liste d'hyperliens vers les adresses des documents jugés pertinents. Cette liste est souvent ordonnée selon la distance de chaque document par rapport à la requête. Toutefois, l'aspect d'interactivité avec un poste informatique connecté à Internet, offre la possibilité d'exploiter l'appréciation préliminaire de l'utilisateur en vue de relancer une nouvelle recherche plus adéquate. Ainsi, la visualisation des résultats de recherche n'est plus considérée comme dernière phase du processus de recherche mais plutôt comme articulation centrale dans un processus interactif mettant en évidence la collaboration du demandeur **[2].**

## I.4. Modèles des systèmes d'indexation et de recherche d'information

La tâche de représentation des documents constitue la plate-forme sur laquelle tout ‹SRI› puisse capturer le contenu documentaire avant de pouvoir l'indexer et de mesurer sa pertinence par rapport à une requête donnée.

L'idée de base, initiée par **[8]** pour indexer le texte en calculant la fréquence de ses termes, est commune pour la plupart des modèles de représentation des documents.

Néanmoins nous pouvons classer ces modèles d'une part selon leurs bases mathématiques : approche de la théorie des ensembles, approche algébrique ou approche probabiliste. D'une autre part, la distinction entre les différents modèles peut être établie sur la base de la prise en compte de l'interdépendance des termes d'indexation. Nous esquissons dans ‹**Figure 3**› les principaux modèles de documents.

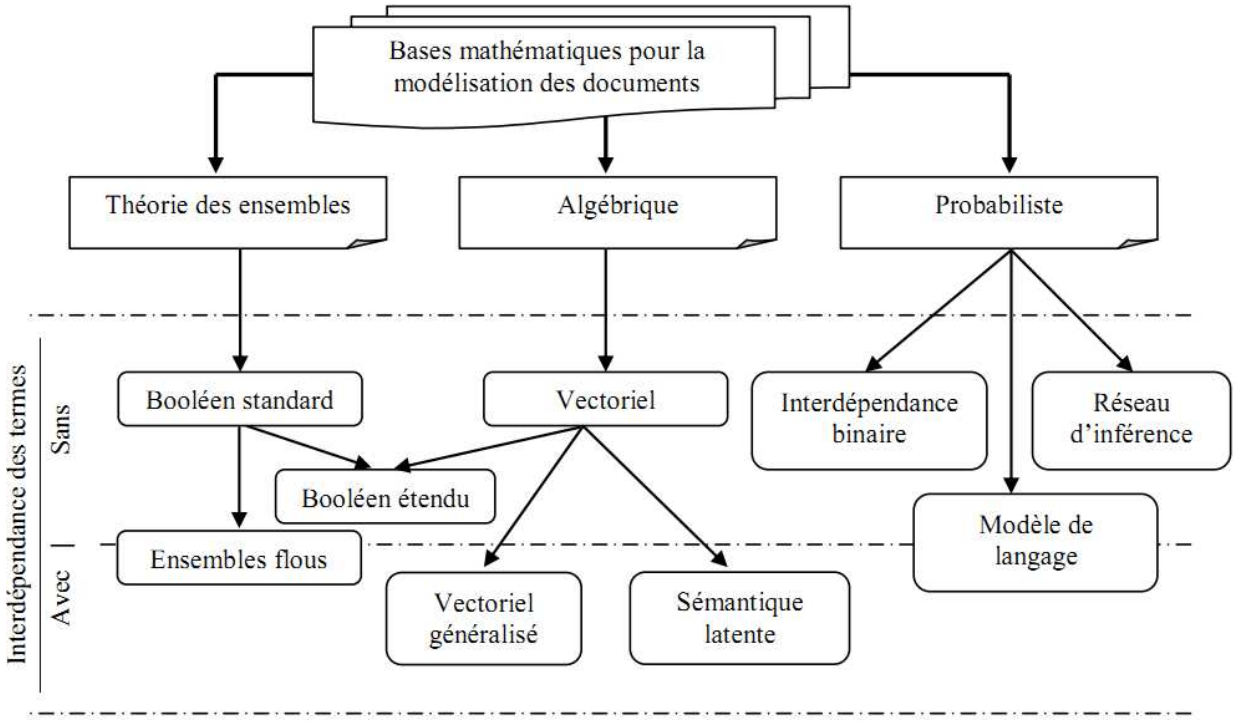



**Figure 3.** Classification des principaux modèles de document pour la RI.

Nous nous contenons dans la suite par les deux modèles qui nous intéresse le plus dans notre travail ; il s'agit des modèles booléen et vectoriel.

### I.4.1. Le Modèle booléen

L'approche Booléenne est la plus ancienne des stratégies de recherche et de modélisation des documents en ﴾RI﴿. Partant d'un modèle ensembliste, l'approche Booléenne caractérise chaque document par l'appartenance ﴾ou non﴿ des termes au même document. L'ensemble de tous les termes utilisés ﴾T﴿ est appelé vocabulaire d'indexation. Ainsi, nous pourrons construire, pour une collection de documents ﴾D﴿, une matrice d'incidence binaire ﴾D,T﴿ où chaque élément ﴾ $d_i, t_j$﴿ prend la valeur 1 si le document di contient au moins une fois le terme tj. Autrement, la valeur 0 interprète l'absence dans le document du terme $d_i$ du terme $t_j$ **[2].**

Prenons, à titre d'exemple, les phrases suivantes :

D1 : L'Algérie compte huit millions d'élèves scolarisés.
D2 : Le vaccin antigrippal est disponible par millions aux Algériens.
D3 : Les algériens ont payé leur indépendance par des millions de martyrs.
D4 : L'équipe nationale est qualifiée à la coupe du monde.
D5 : L'économie nationale dépend fortement des hydrocarbures.

En ne sélectionnant que les noms de ces phrases, nous construisons le vocabulaire d'indexation avec lequel la matrice d'incidence peut être établie comme dans ﴾**Tableau 1**﴿.

**Tableau 1.** Matrice d'incidence ﴾document-terme﴿ selon le modèle booléen.

| | **Algérie** | **algériens** | **millions** | **antigrippal** | **coupe** | **économie** | **équipe** | **élèves** | **hydrocarbures** | **indépendance** | **martyrs** | **monde** | **nationale** | **scolarisés** | **vaccin** | **…** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **$D_1$** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| **$D_2$** | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| **$D_3$** | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | |
| **$D_4$** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| **$D_5$** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | |

### I.4.2. Le Modèle vectoriel

C'est un autre modèle souvent utilisé. Il représente les documents et les requêtes comme vecteurs de poids dans un espace multidimensionnel, dont les dimensions sont les termes utilisés pour construire un index qui représente les documents **[1].**

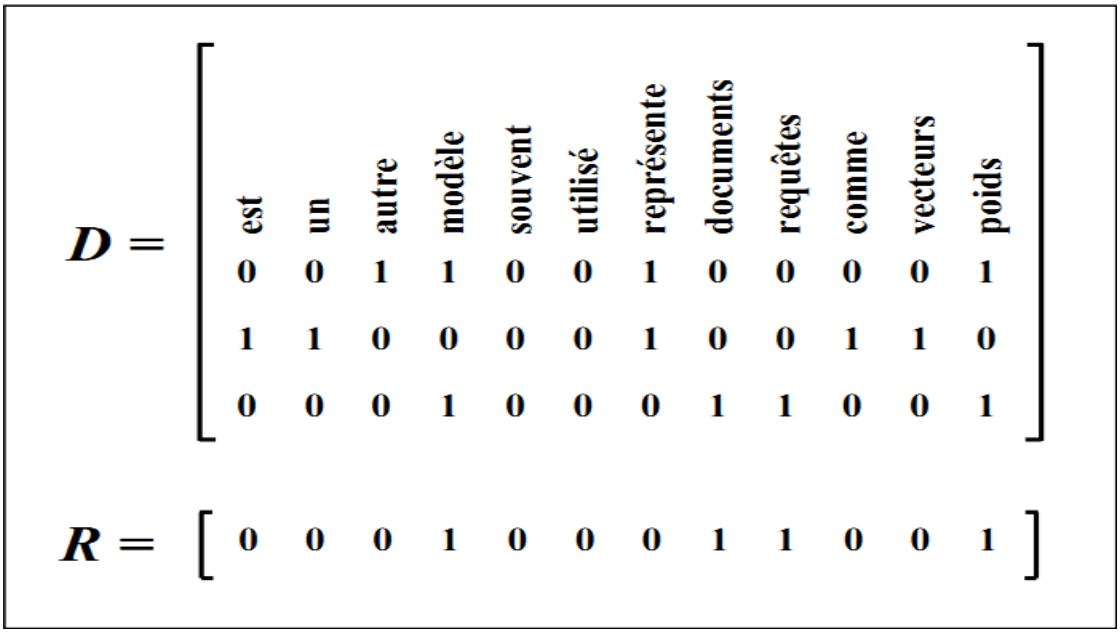



**Figure 4.** Exemple sur la représentation des documents et des requêtes.

La création d'un index implique une lecture lexicologique pour identifier les termes significatifs, où l'analyse morphologique ramène les différentes formes de mot aux « lemmes » communs, et l'occurrence de ces lemmes est calculée. Des substituts de requête et de document sont comparés selon leurs vecteurs. Par exemple, Soit l'espace vectoriel suivant:

$$<t_1, t_2, t_3, \ldots, t_n>$$

Un document et une requête peuvent être représentés comme suit:

$$d = <a_1, a_2, a_3, \ldots, a_n>$$

$$q = <b_1, b_2, b_3, \ldots, b_n>$$

Ainsi, ai et bi correspondent aux poids du terme ti dans le document et dans la requête.

Étant donnés ces deux vecteurs, leur degré de correspondance est déterminé par leur similarité. Il y a plusieurs façons de calculer la similarité entre deux vecteurs. En voici quelques unes:

Sim0 $(d, q) = \Sigma_i (a_i * b_i)$ (produit interne)

Sim1 $(d, q) = \Sigma_i (a_i * b_i) / [\Sigma_i (a_i)^2 * [\Sigma_i (b_i)^2]^{1/2}]$ (cosinus)

Sim2 $(d, q) = 2\, \Sigma_i (a_i * b_i) / [[\Sigma_i (a_i)^2 + [\Sigma_i (b_i)^2]]$

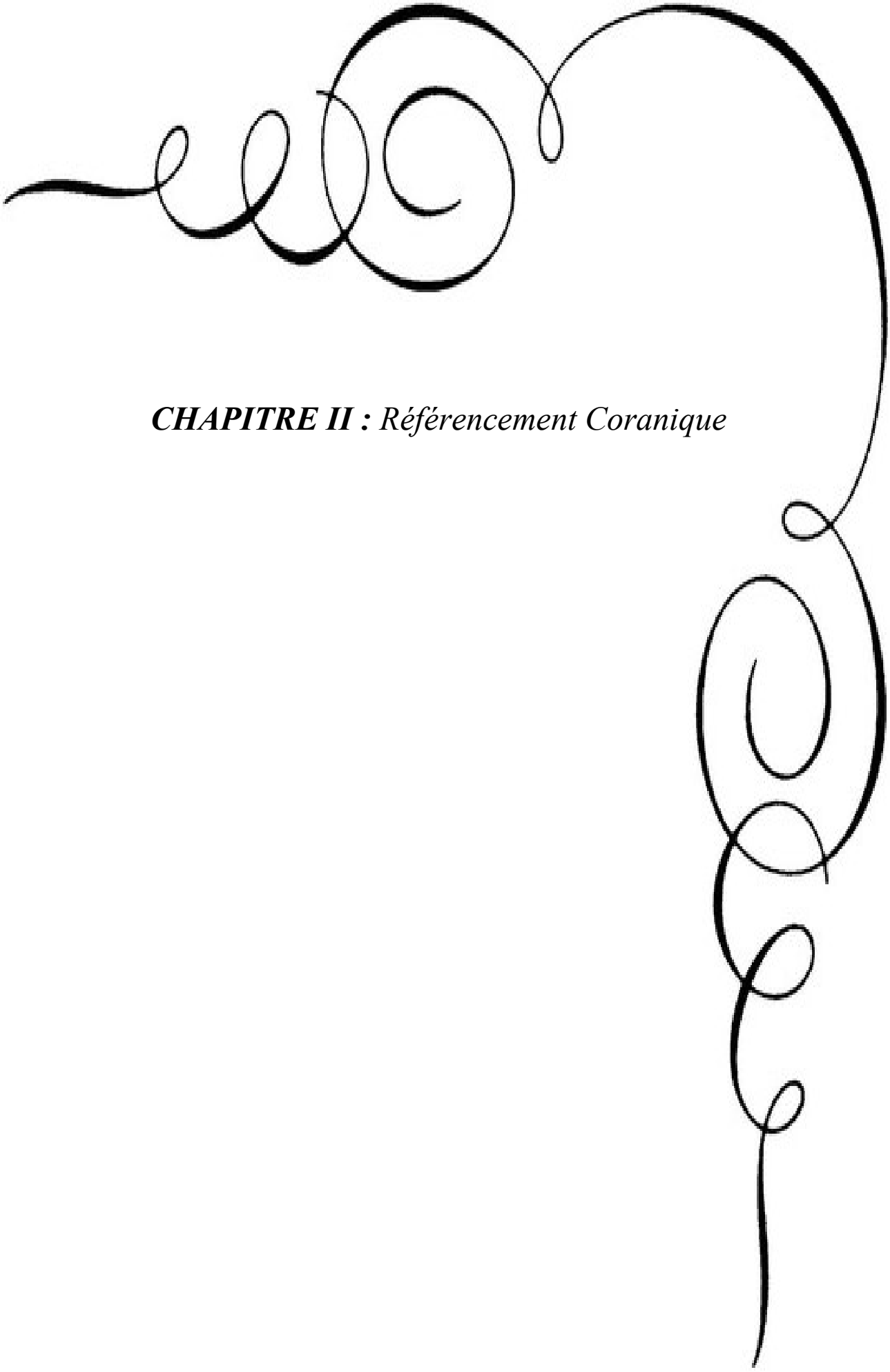
Sim3 $(d, q) = \Sigma_i (a_i * b_i) / [[\Sigma_i (a_i)^2 + [\Sigma_i (b_i)^2 - \Sigma_i (a_i * b_i)]]$

Sauf la première formule, toutes les autres sont normalisées, c'est-à-dire qu'elles donnent une valeur dans [0, 1].

Dans le modèle vectoriel, les termes d'un substitut de requête peuvent être pesés pour tenir compte de leur importance, et ils sont calculés en utilisant les distributions statistiques des termes dans la collection des documents **[1].** Ce modèle peut assigner un haut classement à un document qui contient seulement quelques termes de requête si ces termes se produisent rarement dans la collection mais fréquemment dans le document.

## I.5. Conclusion

Le but de ce chapitre était de présenter les concepts de base de la recherche d'information nécessaires à la formalisation de notre travail. Le modèle joue un rôle central dans la RI. C'est le modèle qui détermine le comportement clé d'un système de RI. Pour répondre à un besoin en information, nous avons décrit le processus standard depuis la collecte et l'indexation des documents jusqu'à l'appariement et la reformulation de la requête de recherche.



## ***CHAPITRE II :*** *Référencement Coranique*

## II.1. Introduction

Par ses propriétés morphologiques et syntaxiques, la langue arabe est considérée comme une langue difficile à maîtriser dans le domaine du traitement automatique de la langue. L'arabe doit sa formidable expansion à partir du $7^{éme}$ siècle grâce à la propagation de l'islam et la diffusion du Coran. Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970**[9].** Les premiers travaux concernaient notamment les lexiques et la morphologie.

Par ailleurs le **Saint Coran** est la parole du Dieu et est écrit en arabe, et donc il représente une ressource fiable pour valider notre approche d'indexation et d'exploration sémantique coranique dans les textes arabes.

Pour arriver à réaliser cette approche, il faut d'abord analyser la nature du texte recherché ﴿les caractéristiques de la langue arabe﴾ pour trouver des solutions aux problèmes d'analyse syntaxique et morphologique, ensuite, il est nécessaire de maitriser les éléments constituant le texte coranique pour arriver à concrétiser la tâche du référencement.

## II.2. La langue arabe

L'Arabe est une langue fortement flexionnelle qui a une structure morphologique complexe. La recherche d'information sur le texte arabe exige la forme de base du mot ﴿racine ou lemme﴾ pour être la plus pertinente. La lemmatisation peut être définie comme un processus qui consiste à retirer tous les affixes ﴿préfixes, infixes, ou/et suffixes﴾ d'un graphème afin de restituer son entrée lexicale unifiée (lemme ou racine). Il est donc nécessaire de connaitre les caractéristiques et les schémas de construction dans cette langue.

### II.2.1. Caractéristiques de la langue arabe

L'alphabet de la langue arabe compte 28 lettres. L'arabe s'écrit et se lit de droite à gauche. Les lettres changent de forme de présentation selon leur position (au début, au milieu ou à la fin du mot) **[9].** Un mot arabe s'écrit avec des consonnes et des voyelles. Les voyelles sont ajoutées au-dessus ou au-dessous des lettres ﴿﴿SUKUN﴾ سكون, ﴿DAMMA﴾ ضمة , ﴿KASRA﴾ كسرة , ﴿FATHA﴾ فتحة ﴾. Elles sont nécessaires à la lecture et à la compréhension correcte d'un texte, elles permettent de différencier des mots ayant la même représentation.

Le lexique arabe comprend trois catégories de mots : verbes, noms et particules. Les verbes et noms sont le plus souvent dérivés d'une racine à trois consonnes radicales.

### II.2.2. Difficultés du traitement automatique de l'arabe

La richesse morphosyntaxique de la langue arabe a suscité des efforts considérables pour l'adaptation des outils de ﴿RI﴾ dans le texte arabe. En effet, ce dernier nécessite un traitement automatique spécifique afin de prendre en charge les difficultés suivantes :

#### II.2.2.1. L'absence de voyelles

Un des aspects complexes de la langue arabe est l'absence de voyelles **[10],** qui risque de générer une certaine ambiguïté à deux niveaux ﴿le sens du mot et l'identification de sa fonction dans la phrase﴾.

Ceci peut influencer les fréquences des mots étant donné qu'elles sont calculées après la détection de la racine ou la lemmatisation des mots qui est basée sur la suppression de préfixes et suffixes. Lors du calcul des scores à partir des titres, il peut arriver que des mots soient considérés comme dérivants d'un même concept alors qu'ils ne le sont pas.

Dans l'exemple 1 **[10],** en utilisant la distribution des mots avec ou sans lemmatisation, la phrase 3 aura un score le plus important alors que les phrases 1 et 2 semblent plus intéressantes, ce qui n'aurait pas été le cas avec un texte voyellé.

**Tableau 2.** Effet du mot non voyellé العلم sur les extraits.

| | |
|---|---|
| العنوان: اثر العلم | **Titre : impact de la science** |
| 1- العلماء..... | **1 - Les scientifiques ….** |
| 2- علميا..... | **2 - Scientifiquement….** |
| 3- في المحاضرة ليس العلم الوطني..... | **3 - A la conférence non seulement le drapeau national… mais aussi le drapeau de chaque pays….** |
| ولكن العلم لكل الدول..... | |

L'ambiguïté vient du mot العلم *la science* ou *drapeau* alors que voyellé on aura العِلْم pour *la science* et العَلَمُ pour *le drapeau*. Cette ambiguïté pourrait, dans certains cas, être levée soit par une analyse plus profonde de la phrase ou des statistiques (par exemple il est plus probable d'avoir العلم الوطني *le drapeau national* que la *science nationale*). De plus la capitalisation n'est pas employée dans l'arabe ce qui rend l'identification des noms propres, des acronymes, et des abréviations encore plus difficile.
Comme la ponctuation est rarement utilisée, on doit ajouter une phase de segmentation de phrase pour l'analyse d'un texte.

### II.2.2.2. L'irrégularité de l'ordre des mots dans la phrase

L'ordre des mots en arabe est relativement libre. D'une manière générale, on met au début de la phrase le mot sur lequel on veut attirer l'attention et l'on termine sur le terme le plus long ou le plus riche en sens ou en sonorité. Cet ordre provoque des ambiguïtés syntaxiques artificielles dans la mesure où il faut prévoir dans la grammaire toutes les règles de combinaisons possibles d'inversion de l'ordre des mots dans la phrase.

### II.2.2.3. Problèmes de segmentation de textes

Pour analyser un texte, nous devons procéder à sa segmentation en paragraphes, phrases et propositions. Cette segmentation est source d'ambiguïtés, vu que d'une part la ponctuation est rarement utilisée dans les textes arabes et d'autre part cette ponctuation, lorsqu'elle existe, n'est pas toujours déterminante pour guider la segmentation.

### II.2.2.4. Problèmes d'agglutination de mots

Contrairement à la plupart des langues latines, en arabe, les articles, les prépositions, les pronoms, etc., se collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent.
Comparé au français, un mot arabe peut parfois correspondre à toute une phrase **[11].** Par exemple, le mot arabe 'أتتذكروننا' [*Ott*k~rwnnA*] correspond en français à la phrase "*Est-ce que vous vous souvenez de nous ?*".

Cette caractéristique engendre des ambiguïtés morphologiques au cours de l'analyse. En effet, il est parfois difficile de distinguer entre un proclitique ou enclitique et un caractère du mot en question. Par exemple, le caractère 'و' [w] dans le mot 'وصل ' [*wSl*] ﴿est *arrivé*﴾ est un caractère qui fait partie de ce mot alors que dans le mot 'و فتح' [*wftH*] ﴿*et a ouvert*﴾, il s'agit d'un proclitique.

Cette analyse comprend "classiquement" les opérations suivantes **:**

- Découper le texte en phrases et segmenter chacune des phrases en séquences d'unités lexicales ﴿mots, expressions, …﴾.
- Déterminer pour chaque unité lexicale, déjà segmentée, ses caractéristiques morphologiques.
- Déterminer comment ces unités lexicales s'articulent les unes avec les autres pour former des groupes syntaxiques.
- Reconnaître les rapports fonctionnels entre les syntagmes qui déterminent la structure sémantique de chaque phrase.
- Interpréter les structures sémantiques par rapport au contexte de l'énoncé et au modèle du discours.

## II.2.3. Analyse du texte arabe

La segmentation ﴿tokenisation﴾ est un processus nécessaire dans le traitement morphologique de la langue. Le but de la segmentation est de diviser un texte en une suite de tokens afin de préparer le traitement morphosyntaxique ﴿rôle et étiquetage morphosyntaxique﴾. Cependant cette phase est souvent intégrée dans un processus plus fin afin de détecter et indexer communément les unités lexicales multiformes mais sémantiquement et morphologiquement semblables.

Afin de fusionner efficacement les mots arabes, le traitement des aspects combinées ﴿de flexions, de dérivation, d'agglutination et non-vocalisation﴾ suit deux approches : soit par stemming léger, en supprimant des affixes communs, soit par analyse morphologique, en cherchant chaque noyau ﴿racine ou lemme﴾ selon un schéma déterminé **[2].**

### II.2.3.1. Stemming léger

L'efficacité de cette approche dépend de la nature morphologique de la langue, du contenu des listes des affixes utilisé et de l'algorithme lui-même. Le stemming du texte arabe nécessite la prise en charge des formes ambigües et agglutinées impliquant plusieurs dérivations morphologiques. L'analyse d'une telle approche peut être revue dans **[12].**

Ce type d'algorithmes peut effectivement traiter une majorité des cas les plus fréquents. Néanmoins, le mot correct peut être perdu dans d'autres situations. Par exemple, dans le mot [wafiy]وَفِي, quelqu'un peut lire deux prépositions agglutinées signifiant "et dans". Mais un autre va le lire comme un seul nom signifiant "fidèle".

### II.2.3.2. Analyse morphologique

Selon le type de sortie désirée, on peut distinguer deux catégories d'analyse morphologique pour le texte arabe : le stemming à base racine et les stemming à base lemme. Le choix entre telle ou telle approche dépend de la nature de la tâche ﴿RI﴾ qui suivra.

Dans la première catégorie, l'algorithme Khoja a été proposé avec une liste de racines et de schémas afin d'extraire d'un mot une racine correcte **[13].** Cet algorithme permet de produire des racines abstraites ce qui réduit considérablement la dimension de l'espace des descripteurs des documents. Cependant, il conduit vers une confusion ennuyante de sens divergents à cause de la non-vocalisation.

En deuxième catégorie, nous trouvons en littérature un ensemble de ressources lexicales, qui ont été développées en 2002 et raffinées en 2004, pour détecter les règles de flexion et de composition dans un mot arabe **[14].** L'analyseur morphologique de Buckwalter a été incorporé plus tard dans le package AraMorph pour la lemmatisation du texte arabe. Plusieurs solutions peuvent être proposées pour chaque mot en entrée.

### II.2.4. Travaux et ressources linguistiques pour l'arabe

Les traitements automatiques des langues naturelles sont basés principalement sur les ressources linguistiques. Ces derniers sont répartis en ressources orales et ressources écrites. Nous nous intéressons dans notre travail à cette deuxième famille qui se compose principalement de :

#### II.2.4.1. KHOJA stemmer

L'algorithme Khoja a été proposé avec une liste de racines et de schémas afin d'extraire d'un mot une racine correcte **[13]**. Cet algorithme permet de produire des racines abstraites ce qui réduit considérablement la dimension de l'espace des descripteurs des documents. Cependant, il conduit vers une confusion ennuyante de sens divergents à cause de la non-vocalisation.

#### II.2.4.2. Arabic Word Net

Basé sur la ressource ontologique de l'anglais WordNet ‹WN›, le navigateur ‹AWN› est une application autonome qui peut être exécuté sur n'importe quel ordinateur qui dispose d'une machine virtuelle Java. Dans son état actuel, ses principales installations comprennent AWN navigation, la recherche de concepts dans AWN, et la mise à jour avec les dernières données AWN des lexicographes.

Cette recherche peut être effectuée en utilisant l'anglais ou l'arabe. En arabe, la recherche peut être réalisée en utilisant soit l'écriture arabe ou avec la translittération de Buckwalter et peut être pour un mot ou une forme de racine, avec l'utilisation facultative de signes diacritiques **[15]**.

#### II.2.4.3. ARAMORPH

*AraMorph* est un portage en *Java* du produit homonyme développé en *Perl* par *Tim Buckwalter* pour le compte du ‹*LDC*› [2]. Le projet inclut des classes *Java* permettant l'analyse morphologique de fichiers textuels en arabe et ce, quel que soit leur encodage. A cet effet, il

[2] Accessible en ligne à l'adresse http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49.

est proposé 3 fichiers de test dans les principaux encodages utilisés pour la langue arabe : UTF-8, ISO-8859-6 et CP1256.Ce projet inclut également des classes compatibles avec l'architecture de **Lucene**[3], ce qui permet l'**analyse**, l'**indexation** et l'**interrogation** de documents en arabe **[16]**.

### II.2.4.4. Alkhalil

C'est un analyseur morphologique fournissant toutes les voyellations possibles de chaque mot du texte pris hors contexte.

#### II.2.4.4.1. Aperçu sur la version 1 de l'analyseur Alkhalil Morpho-Sys1

L'analyseur morphologique Alkhalil Morpho-Sys est considéré comme étant l'un des plus importants analyseurs open sources **[17].** Après l'analyse des mots arabes par Alkhalil Morpho Sys, le système fourni les informations morpho-syntaxiques suivantes :

- Les voyellations possibles du mot.
- Les affixes qui s'ajoutent aux stems.
- Nature du mot ﴿nom ou verbe ou mot outil﴾.
- Le schème ﴿dans le cas des noms et des verbes﴾.
- Le stem.
- La racine ﴿dans le cas des noms et des verbes﴾.
- L'état syntaxique ﴿dans le cas des noms et des verbes﴾.

#### II.2.4.4.2. Aperçu sur la Version 2 de l'Analyseur Alkhalil Morpho-Sys2

Dans cette section, la deuxième version de l'analyseur morphosyntaxique AlKhalil est présentée. Cette version est basée sur la même philosophie que la première version **[18]** mais sa mise à jour a nécessité deux grandes étapes :

- **Ressources Linguistiques**

La 1ère phase consiste en la régénération à nouveaux de la base de données linguistique. Ensuite un arrangement des ressources linguistiques est réalisé en séparant autant que possible les différentes étiquettes ﴿par exemple l'étiquette schèmes et l'étiquette types des mots﴾ et ceci dans le but d'utiliser cette nouvelle base, à des applications qui peuvent utiliser une seule information au lieu de toutes les informations. De même, certaines erreurs dans les listes des schèmes ont été corrigées. Ceci a permis d'améliorer le taux de mot analysé qui est passé de 93% avec la première version à 98.49% pour la 2ème version.

- **Étapes d'analyse**

AlKhalil Morpho-Sys2 utilise les mêmes étapes d'analyse qu'AlKhalil Morpho-Sys1. Le travail dans cette version consistait à faire quelques modifications dans certaines étapes afin d'augmenter le taux des mots analysés et d'améliorer la vitesse d'analyse.

[3] http://lucene.apache.org/

## II.3. Le texte Coranique

**Le saint Coran** est la parole du Dieu, écrit en arabe et suivi par les croyants musulmans**.** Le Coran est composé de 114 sourates ﴿ou chapitres﴾ incluant au total 6236 Ayates ﴿ou versets﴾[4]. Le texte coranique regroupe 77.476 mots à raison de 12.4 mots par verset selon **[19].**

Le Coran est composé de sourates ﴿assimilables à des **chapitres** de longueurs variables﴾ elles-mêmes constituées de versets ﴿assimilables à des **paragraphes** de longueurs variables﴾. Au sein d'une sourate, les versets sont numérotés du premier jusqu'au dernier.

Pour des raisons de commodité, un système de **découpage** du Coran – en parties de longueurs sensiblement égales – a été mis au point **[20]:**

**Manzil** ﴿منزل, station﴾ : unité ayant pour valeur le 1/7 e de Coran. A raison d'un manzil par jour, on peut lire ou réciter tout le Coran en une semaine.

**Ğuzʾ** ﴿جزء, fraction﴾: unité représentant le 1/30 e du Coran. A raison d'un ğuzʾ par jour, on peut lire ou réciter tout le Coran en un mois.

**Ḥizb** ﴿حزب, section﴾ : unité valant le 1/60 e du Coran. C'est donc la moitié d'un ğuzʾ.

**Niṣf** ﴿نصف, moitié﴾ : il s'agit de la moitié d'un ḥizb, donc du 1/120 e du Coran.

**Rubʿ** ﴿ربع, quart﴾ : unité qui vaut le quart d'un ḥizb, donc le 1/240 e du Coran.

**Ṯumun** ﴿ثمن, un huitième﴾ : unité qui représente le 1/8 e d'un ḥizb, donc le 1/480 e du Coran.

Ce découpage est généralement indiqué sur les **marges** du Coran.

Notons que ce système de division n'était pas d'usage du temps du Prophète de l'islam. **Il est apparu ultérieurement** pour faciliter la récitation, l'apprentissage par cœur et la révision ﴿des passages mémorisés﴾ du texte coranique.

### II.3.1. Historique de l'écriture du Coran

Depuis la révélation du saint Coran au prophète Mohammed, plusieurs améliorations avaient été réalisées pour la sauvegarde l'homogénéité et la clarté du texte coranique. Plusieurs générations ﴿ou modes d'écriture﴾ du "Mushaf" avaient vu le jour **[21].**

#### II.3.1.1. Première étape ﴿simple calligraphie﴾

A l'époque du prophète Mohamed et jusqu'à l'unification du Mushaf à l'époque du troisième Khalifa de l'Islam ﴿Othman﴾, l'écriture du Coran se contentait de la simple calligraphie sans aucune ponctuation ni vocalisation ﴿**Figure 5**﴾.

[4] En considérant seulement "Bismillah …" du premier chapitre "Al-Fatiha" comme verset.

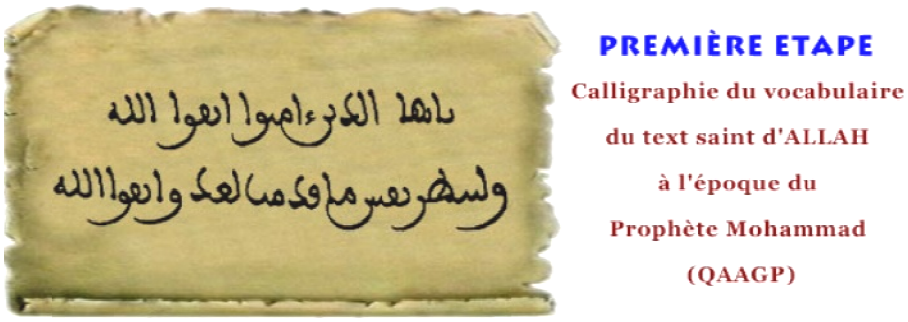



**Figure 5.** Prototype d'un texte coranique de la première étape.

### II.3.1.2. Deuxième étape ﴿vocalisation﴾

A l'époque du quatrième Khalifa de l'Islam ﴿Ali﴾, le texte coranique avait était enrichi par les symboles diacritiques afin de le vocaliser complètement et enlever l'ambigüité morphosyntaxique chez les nouveaux non-arabes convertis à l'Islam ﴿**Figure 6**﴾.


**Figure 6.** Prototype d'un texte coranique de la deuxième étape.

### II.3.1.3. Troisième étape ﴿ponctuation﴾

Pour la même raison, et durant l'époque du Khalifa AbdelMalik Ibn Marwan, le texte coranique avait connu l'introduction de ponctuation distinguant complètement les lettres arabes ﴿**Figure 7**﴾.


**Figure 7.** Prototype d'un texte coranique de la troisième étape.

### II.3.1.4. Quatrième étape ‹codification colorée›

Vers la fin des années 1990, une nouvelle version du Mushaf avait vu le jour ‹Mushaf Tajwid›. Elle incluait une transcription de codification colorée aidant à l'apprentissage de la bonne prononciation ‹**Figure 8**›.


**Figure 8.** Prototype d'un texte coranique de la quatrième étape.

### II.3.2. Statistiques du texte coranique

Certaines analyses statistiques du texte coraniques ont été réalisées de façon manuelle, actuellement la plupart repose sur les versions électroniques, mais les résultats annoncées ne sont pas unifiés et parfois contestable; ceci revient surtout aux considérations diversifié de l'unité lexicale elle-même :

- Version de référence ‹Hafs, Warch, …›,
- Lettre écrite et lettre prononcée,
- Premier "Bismillah" du chapitre comme verset ou non,
- Interprétations morphosyntaxiques différentes du graphème coranique,
- Lemmatisation ou segmentation adoptée.

Néanmoins, ceci n'exclut pas que ces différences restent minimes et donnent pratiquement les mêmes rapports de cooccurrences.

Parmi les 77.476 mots, il n'existe que 17.622 mots coraniques distincts[5] en respectant la distinction par vocalisation. 9.089 mots sont cités une seule fois, alors que 2.399 se répètent deux fois; le tableau 3 en résume quelques statistiques **[19]:**

**Tableau 3.** Quelques statistiques de l'analyse d'un corpus électronique du coran.

| | |
|---|---|
| **Nombre de mots** | **17.622** |
| **Nombre d'occurrences** | 77.476 |
| **Rapport occurrences/formes** | 4,4 |
| **Mots uniques** | 9.089 |
| **Mots doubles** | 2.399 |
| **Mots triples** | 1.036 |

[5] En considérant distincts le mot avec ou sans proclitique "w, و".

### II.3.2. Travaux et ressources pour le Coran

#### II.3.2.1. Le projet Tanzil

**Tanzil** est un projet visant à fournir du Coran, un texte de Coran précis très vérifié.

La première version du texte Tanzil Coran est publiée en Janvier 2007 **[22].**

#### II.3.2.2. Pourquoi Tanzil ?

Depuis l'apparition de la première copie numérique du coran, il y a eu un effort considérable pour produire un texte numérique précis du Coran, mais en raison de certaines difficultés comme le manque des signes diacritiques, ces efforts ont échoué dans la plupart des cas, et malheureusement, les textes du coran sont apparus dans la majorité des sites web et des applications coraniques souffraient de beaucoup d'erreurs et fautes de frappes.

Cette situation terrible a motivé de lancer le projet Tanzil, de produire un texte de Coran sans erreur très vérifié, et faire de ce texte une source disponible pour les sites Web du Coran et des applications pour limiter la propagation ultérieure des textes erronées du Coran.

#### II.3.2.3. Précision du texte coranique Tanzil

Le projet Tanzil a intégré plusieurs travaux antérieurs réalisés dans un certain nombre de projets coraniques pour obtenir un texte unique Coran, et a introduit le texte à un haut niveau de précision en passant par plusieurs phases manuelles et automatiques de vérification. Le long travail qui a conduit au texte final Tanzil est résumé dans les trois étapes suivantes:

##### II.3.2.3.1. Extraction de texte automatique

À cette étape, un recueil de textes du Coran de plusieurs ressources authentiques a été recueilli, et un programme a été élaboré pour convertir chaque texte à un format canonique. Les textes canoniques ont ensuite été analysés et comparées avec soin pour en extraire un texte coranique de base.

##### II.3.2.3.2. Vérification à base de règle

Dans cette étape, un programme a été élaboré pour vérifier la base du texte du Coran contre un ensemble de règles grammaticales et de récitation. Le programme a également été en mesure de produire des textes du Coran dans deux formats texte arabe standard ﴿Imlaei﴾ et texte coranique original ﴿Uthmani﴾, les deux dérivés automatiquement à partir du texte de base.

##### II.3.2.3.3. Vérification manuelle

Dans cette étape, le texte coranique produit a été minutieusement examiné par rapport au Mushaf de Medina, avec une série de contrôle pour chaque lettre et diacritique. L'équipe de développement avait largement utilisé l'aide d'un groupe d'experts comprenant des spécialistes du Coran et Hafizes afin d'assurer l'exactitude et la précision du texte de Coran obtenu.

Ce texte Coran est maintenant utilisé dans les grands sites coraniques et des projets, avec plusieurs millions d'utilisateurs actifs par mois. Malgré ce volume élevé d'utilisateurs, aucune erreur de frappe n'est détectée depuis la sortie de la première version du texte en

2008. La précision du texte a également été confirmée par plusieurs projets qui utilisent activement le texte coranique Tanzil.

### II.3.2.4. Applications basées sur Tanzil

Depuis l'apparition du projet Tanzil, diverses applications, exploitant cette ressource ouverte, ont vu le jour. Nous citons quelques projets de référence :

#### II.3.2.4.1. JQuranTree

*JQuranTree* est un ensemble de packages Java pour l'accès et l'analyse du texte coranique basée sur Tanzil. Il est organisé en trois parties**:** Le texte coranique lui-même, une bibliothèque de fonctions d'accès et une autre pour l'analyse. La distinction entre l'accès et l'analyse du Coran est que les fonctions d'accès se préoccupent de représenter le texte arabe ﴾par exemple, des chapitres, des versets, des lettres et des signes diacritiques﴿, tandis que l'API d'analyse est construite au-dessus de celle-ci en fournissant des outils plus sophistiqués pour la linguistique computationnelle.

#### II.3.2.4.2. Le corpus Coranique arabe

Une ressource linguistique annoté qui montre la grammaire arabe, la syntaxe et la morphologie de chaque mot dans le Coran. Le corpus fournit trois niveaux d'analyse: une annotation morphologique, un Treebank syntaxique et une ontologie sémantique. Ce projet contribue à la recherche du Coran en appliquant la technologie de l'informatique en langage naturel pour analyser le texte arabe de chaque verset **[23].**

#### II.3.2.4.2.1. L'arbre de dépendance coranique

Le Treebank coranique est un effort pour projeter l'ensemble de la grammaire du Coran par en liant les mots selon leur dépendances. La structure linguistique des versets est représentée en utilisant la théorie des graphes mathématique. Le corpus annoté fournit une nouvelle visualisation de la syntaxe du Coran à l'aide de graphes de dépendance.

#### II.3.2.4.2. L'ontologie de concepts coraniques

L'ontologie coranique utilise la représentation des connaissances pour définir les concepts clés dans le Coran, et montre les relations entre ces concepts en utilisant la logique des prédicats. Les entités nommées dans les versets, comme les noms, les personnes historiques et lieux mentionnés dans le Coran, sont liés à des concepts dans l'ontologie.

#### II.3.2.4.1.3. Autres application

Sans description détaillée, nous présentons ci-dessous, d'autres applications exploitant la même ressource Tanzil[6] :

### Sites Web

- **Muslim Web**[7] : Un site qui offre la possibilité de parcourir les chapitres et les versets du Coran et les liants avec différentes traductions et explications et récitations variées.

[6] Voir détail sur la page http://tanzil.net/wiki/Who_is_using_Tanzil?
[7] http://quran.muslim-web.com

- **Search the Quran**[8] : Un moteur de recherche dans le texte coranique offrant diverses possibilités pour l'analyse de la requête de recherche.

### Applications de bureau

- **Zekr Qur'an**[9] : une application coranique open-source pour Windows, Mac et Linux. L'épine dorsale de Zekr est très générique, permettant la personnalisation de plusieurs façons. Personnaliser la langue, la traduction, la récitation, commentaire, et le thème.
- **Quran with Tafseer**[10] : une application coranique pour Windows en ourdou et en anglais.
- **Dangi Coran**[11] : un logiciel pour Windows coranique en kurde, arabe et anglais.
- **Noor**[12] : un petit afficheur du Coran en Python.
- **Bot Coran**[13] : un robot client du Jabber (un système standard et ouvert de messagerie instantanée) qui affiche du texte du Coran dans différents formats de texte. Actuellement, le site propose tout un système d'exploitation intégrant plusieurs applications coraniques.
- **Alfanous**[14] : un moteur de recherche coranique pour Windows et Linux.

### Applications pour mobile

- **alQuran**[15] : pour IPhone et IPod
- **uQuran**[16] : pour téléphones mobiles (en général)
- **Qiraat**[17] : pour Android.

## II.4. Exploration thématique du Coran

Dar Al-Maarifah en Syrie avait obtenu le brevet scientifique pour son innovation du "Mushaf Al-Tajwid" en 1994. La première reconnaissance légale a été obtenue en 1999 de la part de l'université d'Al-Azhar en Egypte. Le produit propagé partout ailleurs à travers le monde; il incluait, entre autre, une arborescence thématique du Coran. Le "Mushaf Al-Tajwid" contient un index hiérarchique complet ou une ontologie de près de 1200 concepts dans le Coran **[21].**

Importé de "Mushaf Al-Tajwid", "Quran-Topics" est un outil en-ligne qui couvre tous les thèmes et concepts mentionnés dans le Coran **[24].** Les Chercheurs peuvent utiliser le navigateur de l'ontologie Quran-Topics, d'identifier un concept précis et trouver les versets qui font allusion à ce concept, avec plus de précision. Toutes les traductions anglaises pour les

[8] http://search-the-quran.com/
[9] zekr.org
[10] http://www.ecrore.com/mkashif/qwt/qwt.html
[11] http://www.dangiislam.org/download/111
[12] http://noor.sourceforge.net/
[13] http://sabily.org/website/
[14] http://cms.alfanous.org/
[15] http://iphone.almubin.com/alQuran/
[16] www.guidedways.com/mobile/uquran/
[17] https://play.google.com/store/apps/details?id=org.qiraat

concepts sont obtenues à partir de la traduction anglaise de "Mushaf Al-Tajwid". L'explorateur "Quran-Topics" est optimisé pour la recherche rapide des versets.

## II.5. Référencement

### II.5.1. Référencement bibliographique manuel

Une des fonctions manuelles qu'un écrivain doit accomplir est de référencer ses propos. L'objectif est de les appuyer par d'autres récits ou de citer leurs sources bibliographiques.

#### II.5.1.1. Définitions

Dans une publication scientifique **[25]:**

- Une référence bibliographique est l'ensemble des éléments qui décrivent un document et permettent de l'identifier ;
- Une citation bibliographique est une référence brève à un document, placée dans le corps du texte de la publication ;
- L'ensemble des références bibliographiques des documents utilisés lus, ou du moins consultés, pour rédiger une publication donnée forme la liste des références bibliographiques ;
- Une bibliographie est une liste autonome de documents, consultés ou non, relatifs à un sujet donné.

#### II.5.1.2. Pourquoi est-il important?

Dans les pratiques de rédaction scientifique, il est commode d'inclure les références bibliographiques dans la mesure où : **[25]**

- Est cruciale pour une recherche réussie.
- Aide les lecteurs à trouver la source d'origine s'ils le souhaitent.
- Améliore vos compétences en écriture
- Ajoute une authenticité à votre argument.
- Indique que vous avez lu.
- Peut vous aider à obtenir de meilleures notes.

### II.5.2. Référencement automatique

Parmi les tâches de recherche d'information on peut citer la catégorisation des textes, le clustering, le filtrage et le résumé automatique. La recherche ad-hoc (par requête libre) reste de loin le noyau de ces tâches puisqu'elle regroupe l'essentiel des fonctionnalités d'analyse textuelle telles que l'indexation, l'organisation, la représentation, l'appariement, …etc.

Les autres tâches d'extraction des connaissances (entité, relation, concepts, …) sont affiliés plutôt à au text-mining où elles se greffent comme fonctionnalité supérieure de la recherche d'information en termes d'analyse de document. Par ailleurs, il est intéressant de tracer des associations linguistiques et sémantiques entre les documents. Cette association varie de la simple ressemblance terminologique jusqu'à la dépendance conceptuelle.

Dans ce contexte, on peut invoquer la notion de référencement textuel qui englobe, entre autre, les fonctions suivantes :

- Citation ou référence bibliographique.
- Renvoi de note comme mention marginale.
- Recherche de documents partiellement similaires.
- Détection de plagiat dans les documents scientifiques.

Grâce aux avancées technologiques dans l'analyse automatique des documents, il paraît possible et ambitieux d'automatiser ces tâches naturellement affiliées à la recherche d'information.

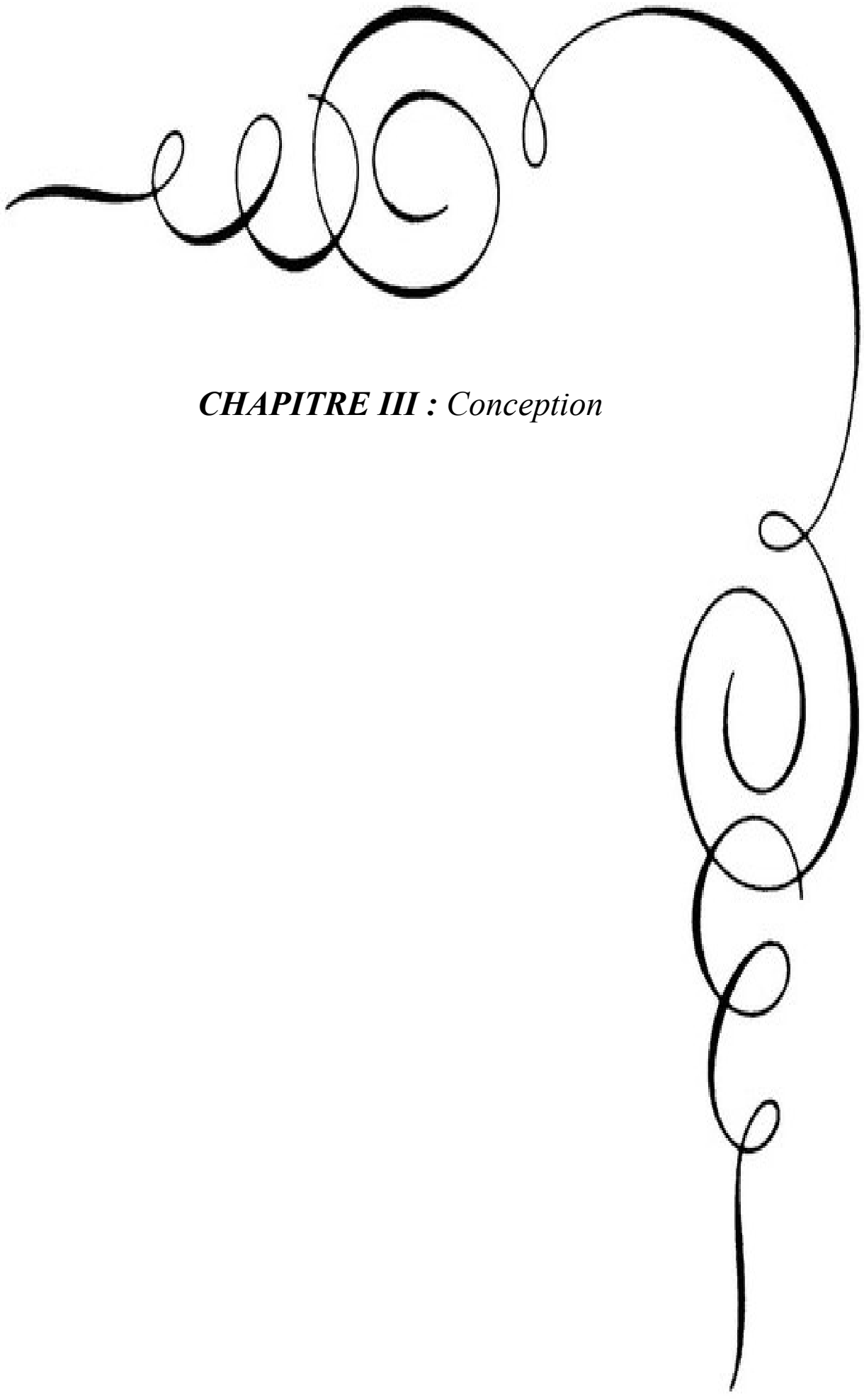
Le Coran se distingue par le fait qu'il reste éternellement une source incontestable pour la langue arabe et pour les différents écrits religieux. Lorsqu'un spécialiste, ou même un simple utilisateur, explore ou rédige un texte arabe, il souhaite souvent trouver une citation coranique pour :

- appuyer ses propos,
- vérifier ces jugements,
- chercher des correspondances terminologiques ou,
- chercher d'autres contextes pour le même texte.

Ces tâches relèvent certes de l'intelligence humaine, mais peuvent faire l'objet de plusieurs sujets d'étude et ouvrent de nouveaux perspectives pour l'exploitation des techniques de la RI et du text-mining au service de la promotion de la rédaction et de la lecture arabe.

## II.6. Conclusion

Au terme de ce chapitre, nous avons vu les caractéristiques de la langue arabe, ainsi que les travaux relatifs à l'automatisation de son analyse. Un aperçu général sur la recherche dans le texte coranique a été exposé. L'utilisation des nouvelles technologies dans la structuration, l'indexation, l'accès et l'analyse du Coran, présente un défi réel. Le projet Tanzil a permis de fournir une ressource électronique ouverte mais surtout fiable et accréditée. L'idée clé du référencement automatique a été décrite dans l'objectif ambitieux de promouvoir la lecture et la rédaction arabe appuyée par les versets du saint Coran.



# ***CHAPITRE III :*** *Conception*

## III.1. Introduction

Après avoir pris connaissance dans les chapitres précédents des méthodes de recherche d'information ainsi que les modèles d'indexation et des problèmes liés à la langue arabe et de ses propriétés morphologiques, ainsi que la nature du texte coranique, on décrit dans ce qui suit les aspects de conception de notre approche et l'application associée.

Ce chapitre présente l'architecture et les différents diagrammes relatifs à la conception de notre système d'indexation, recherche et d'exploration coranique dans les textes arabes.

## III.2. Les diagrammes de modélisation

### III.2.1. Schéma de la base ‹MCD›

Les éléments du texte coranique ont été extraits et structurés dans une base de données relationnelle. Le modèle conceptuel de notre base de données *Quran* est présenté dans ‹Figure 9› :

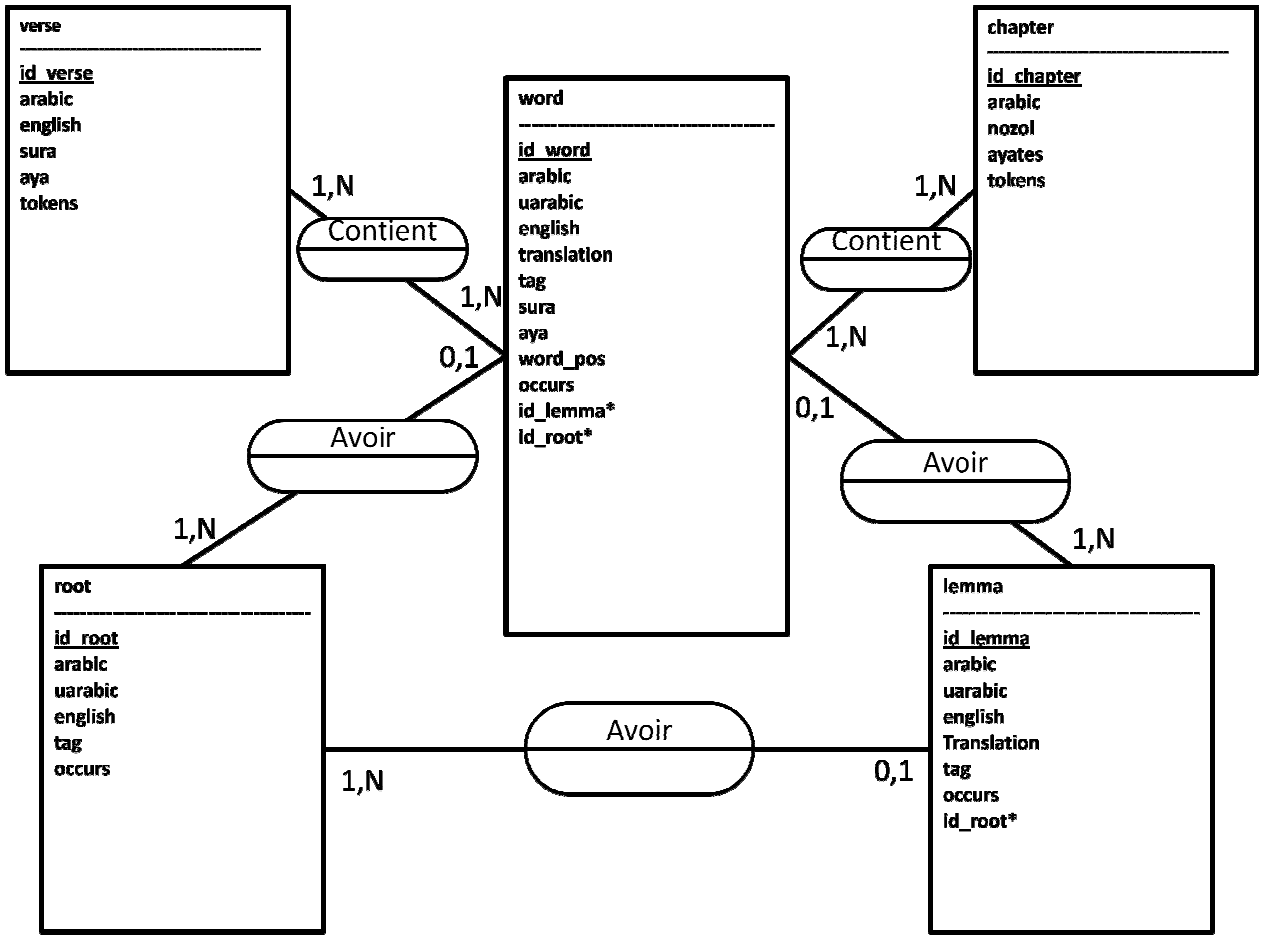



**Figure 9.** Schéma de la base ‹MCD›.

On cite ci-dessous quelques informations qui découlent des principes de la modélisation entité/relation :

0,1 : présence 0 ou une fois au maximum.

0,n : présence de 0 à n fois.

1,1 : présence obligatoire et seulement une fois.

1,n : présence obligatoire en n fois.

Cela peut se comprendre ainsi entre les tables de notre base de données :

***Word et lemma :***

1- Chaque lemme appartient obligatoirement à un mot comme il peut appartenir à plusieurs.
2- Chaque mot n'a pas obligatoirement de lemme et il ne peut en avoir qu'un seul.

***Word et root :***

1- Chaque racine appartient obligatoirement à un mot comme elle peut appartenir à plusieurs.
2- Chaque mot n'a pas obligatoirement une racine et il ne peut en avoir qu'une seule.

***Word et verse :***

1- Chaque mot est associé obligatoirement à un verset comme il peut être associé à plusieurs.
2- Chaque verset contient obligatoirement un mot comme il peut contenir plusieurs.

***Word et chapter :***

1- Chaque mot est associé obligatoirement à un chapitre comme il peut être associé à plusieurs.
2- Chaque chapitre contient obligatoirement un mot comme il peut contenir plusieurs.

***Lemma et root :***

1- Chaque lemme n'a pas obligatoirement une racine et il ne peut en avoir qu'un seul.
2- Chaque racine est associée obligatoirement à un lemme comme elle peut être associée à plusieurs.

On peut donc décrire les tables relationnelles selon le tableau 4 ci-dessous :

**Tableau 4.** Description des tables de la base Quran.

| word | | |
|---|---|---|
| **Nom du champ** | **Type** | **Description** |
| **id_word** | Int (11) | Identificateur du mot |
| **arabic** | Varchar(255) | transcription othmani vocalisée |
| **uarabic** | Varchar(255) | Transcription othmani non vocalisée |
| **english** | Varchar(255) | Translitération buckwalter |
| **translation** | Varchar(255) | Glossaire anglais |
| **tag** | Varchar(255) | Rôle ou catégorie |
| **sura** | Int (11) | Le chapitre contenant le mot |
| **aya** | Int (11) | Le verset contenant le mot |
| **word_pos** | Int (11) | Position du mot dans le verset |
| **occurs** | Int (11) | Nombre d'occurrences |
| **id_lemma** | Int (11) | Identificateur du lemme |
| **id_root** | Int (11) | Identificateur de la racine |

| Lemma | | |
|---|---|---|
| **Nom du champ** | **Type** | **Description** |
| **id_lemma** | Int (11) | Identificateur du lemme |
| **arabic** | Varchar(255) | transcription othmani vocalisée |
| **uarabic** | Varchar(255) | Transcription othmani non vocalisée |
| **english** | Varchar(255) | Translitération buckwalter |
| **translation** | Varchar(255) | Glossaire anglais |
| **tag** | Varchar(15) | Rôle ou catégorie |
| **occurs** | Int (11) | Nombre d'occurrences |
| **id_root** | Int (11) | Identificateur de la racine |

| Root | | |
|---|---|---|
| **Nom du champ** | **Type** | **Description** |
| **id_root** | Int (11) | Identificateur de la racine |
| **arabic** | Varchar(255) | transcription othmani vocalisée |
| **uarabic** | Varchar(255) | Transcription othmani non vocalisée |
| **english** | Varchar(255) | Translitération buckwalter |
| **tag** | Varchar(15) | Rôle ou catégorie de |
| **occurs** | Int (11) | Nombre d'occurrences |

| Chapter | | |
|---|---|---|
| **Nom du champ** | **Type** | **Description** |
| **id_chapter** | Int (11) | Identificateur du chapitre |
| **arabic** | Varchar(255) | transcription othmani vocalisée |
| **nozol** | Varchar(25) | Lieu de révélation du coran |
| **ayates** | Int (11) | Nombre des versets du chapitre |
| **tokens** | Int (11) | Nombre des mots du chapitre |

| Verse | | |
|---|---|---|
| **Nom du champ** | **Type** | **Description** |
| **id_verse** | Int (11) | Identificateur du verset |
| **arabic** | Varchar(1200) | transcription othmani vocalisée |
| **english** | Varchar(1200) | Translitération buckwalter |
| **sura** | Int (11) | Le numéro du chapitre contenant le verset |
| **aya** | Int (11) | Le numéro du verset dans le chapitre |
| **tokens** | Int (11) | Nombre des mots du verset |

### III.2.2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation résume les fonctionnalités principales offertes à l'utilisateur ‹Figure 10› :

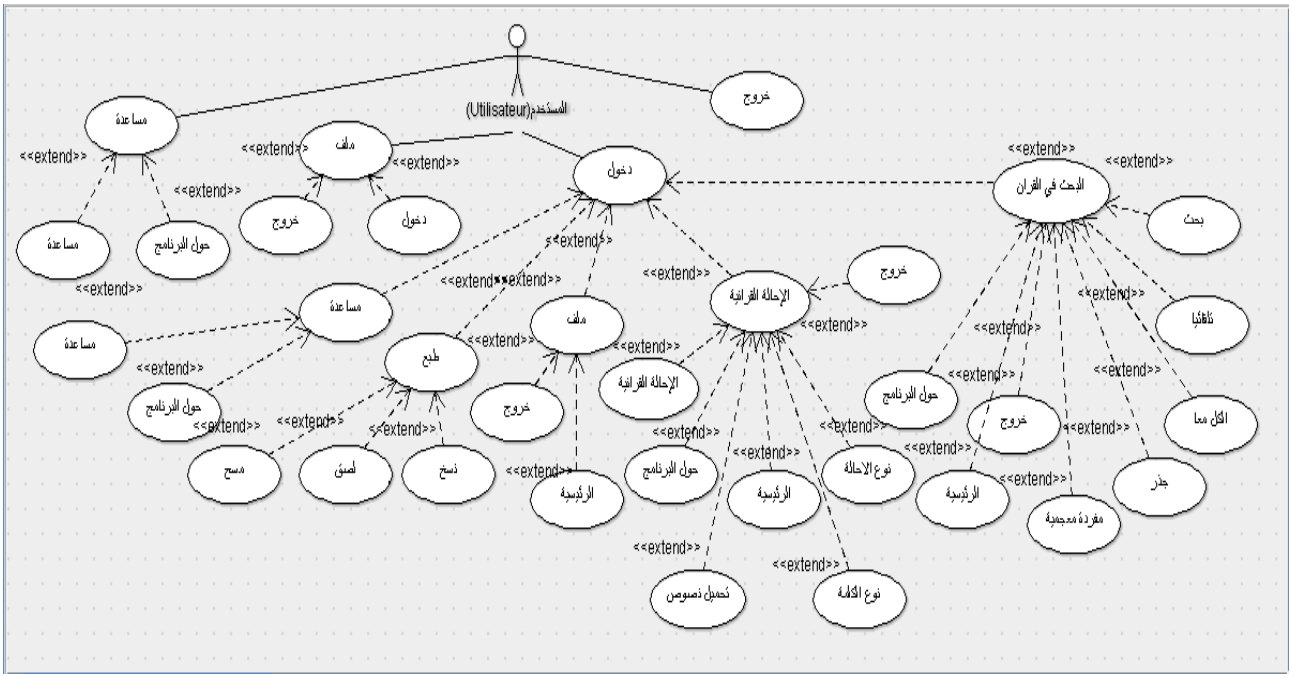



**Figure 10.** Diagramme de cas d'utilisation.

### III.2.3. Architecture de l'application

Nous présentons dans ce qui suit l'architecture globale adoptée pour notre application ﴿Figure 11﴾ :

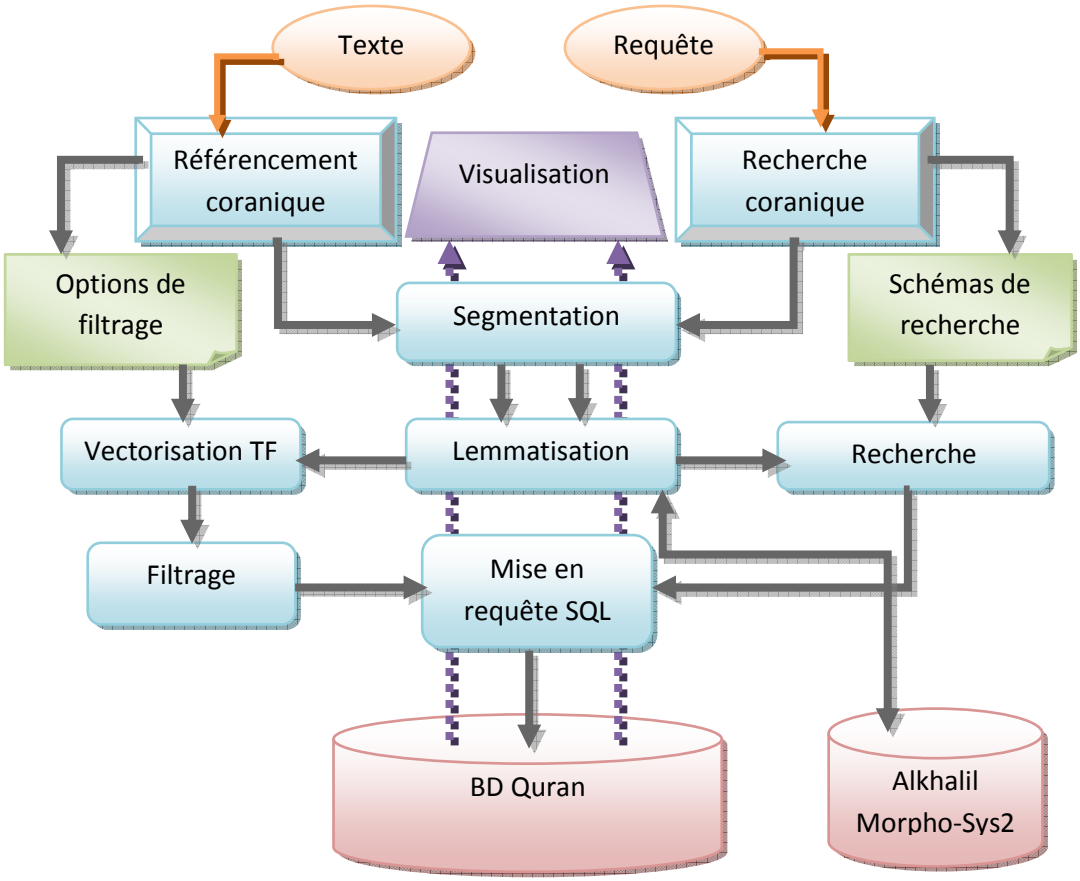



**Figure 11.** Architecture de l'application.

### III.2.4. Diagramme de classes

Nous formalisons la conception de notre travail selon le diagramme de classe suivant ﴿Figure 12﴾ :

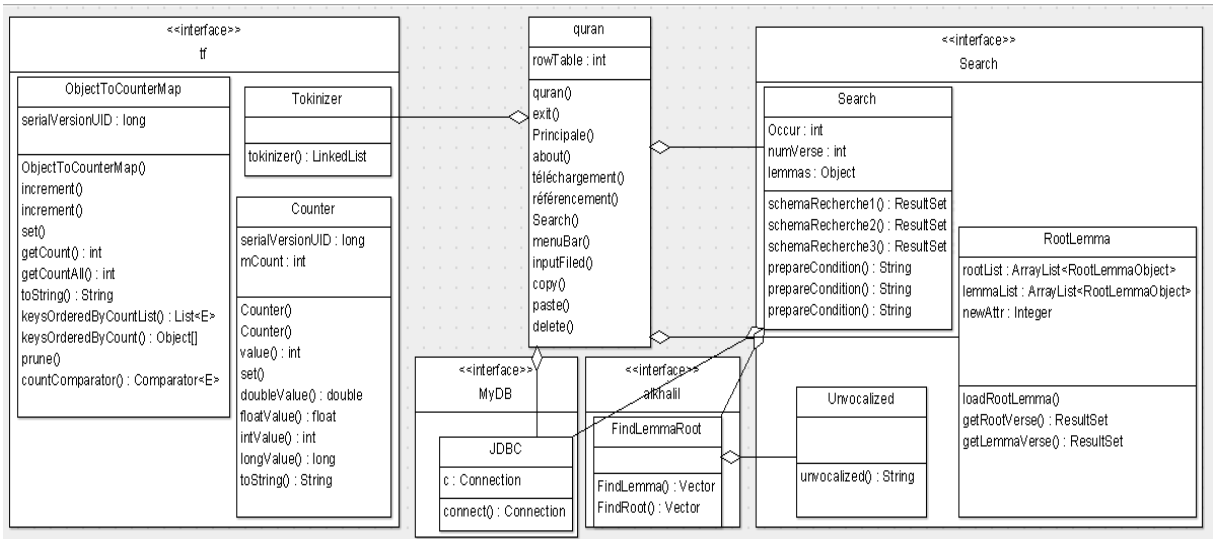



**Figure 12.** Diagramme de classes.

## III.2.5. Processus de référencement

On présente ci-dessous notre approche de référencement basée sur l'appariement linguistique entre le texte arabe et les unités coranique considérées.

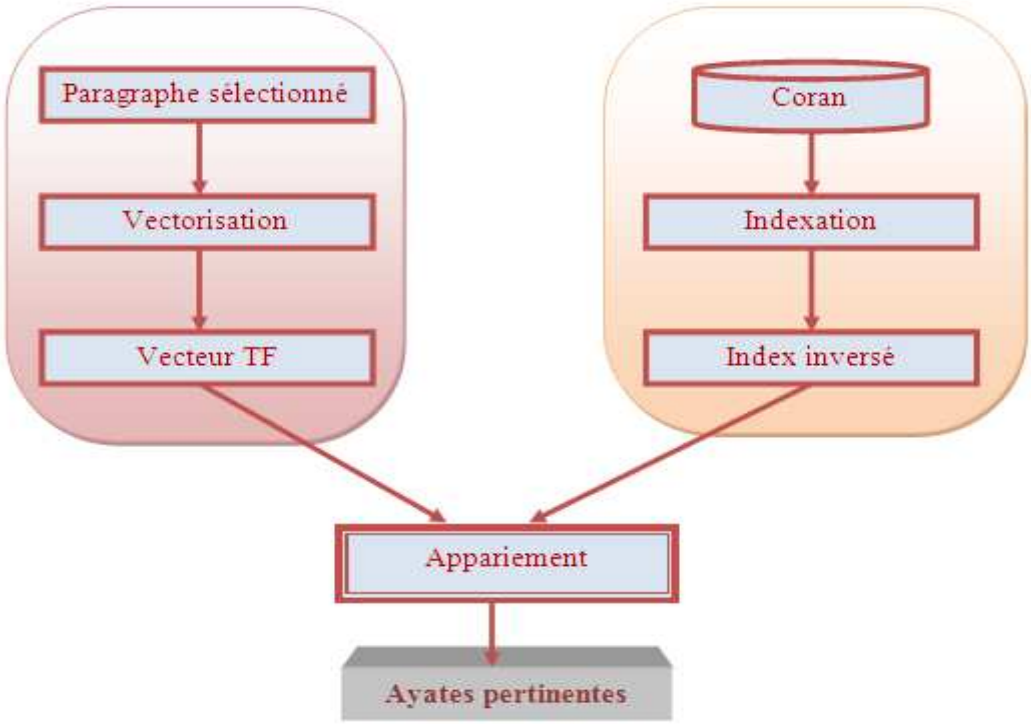



**Figure 13.** Modèle de référencement.

### III.2.5.1. Indexation du Coran

La recherche d'un mot dans des dizaines de documents pouvait prendre plusieurs minutes. Imaginons maintenant une recherche sur le web parmi 8 milliards de pages web !
Il faut donc une structure de données particulière appelée « index inversé » qui, pour un mot donné, nous donne directement la liste des documents où il apparaît, et ce, très rapidement.

L'idée n'est pas nouvelle et on utilise depuis longtemps des concordances en littérature, c'est-à-dire l'ensemble des passages d'un texte où figure un terme. Au début du vingtième siècle, établir une table de concordance d'un roman était un travail qui pouvait prendre des mois d'effort. En comparaison, nous allons voir qu'un moteur de recherche comme **Lucene** peut faire un travail comparable en quelques secondes !

### Index

Dans sa forme la plus simple, un index est tout simplement une structure qui nous donne, pour chaque mot trouvé dans un corpus, la liste des documents où il se trouve.

Un index inversé peut prendre plusieurs formes. Certains index peuvent ne donner que la liste des documents où les mots apparaissent, d'autres vont aussi donner la position des mots dans les documents. Certains index vont effectuer préalablement la troncature des mots, remplaçant *trouvent* par *trouve*, mais d'autres non. La casse des mots doit aussi être traitée, etc.

Pour arriver à réaliser notre index inversé il fallait d'abord passer par une étape de découpage du coran.

### Choix du découpage ﴿Pruning﴾

On peut opter par un découpage en quarts du coran, mais dans ce cas le résultat retourné sera énorme par rapport au besoin de l'utilisateur, et en même temps il n'aura aucun sens, car l'utilisateur sera devant une grande quantité d'informations, et ça va le perturber plus que l'aider. Pour cela on peut penser à un découpage en huitièmes, mais il apparait bien qu'il s'agit du même problème car le huitième est volumineux, et il ne répond pas pertinemment à une requête d'un utilisateur.

On est obligé maintenant de trouver une autre façon pour découper le Coran en petites unités portants un sens et améliorant la pertinence des résultats retournés, les chapitres sont moins volumineux que les deux structures précédentes et ils portent un sens, mais le problème est qu'il y a des chapitres qui sont tellement volumineux ﴿El baquara , البقرة﴾ qu'ils peuvent engendrer un problème de perturbation pour l'utilisateur, et donc cette structure est valide pour quelques chapitres mais pas tous le coran. La seule structure qui donne des résultats pertinents et qui palie tous les problèmes cités précédemment est le découpage en versets, car c'est la dernière unité de découpage portant un sens et en même temps c'est la moins volumineuse est donc elle représente la meilleure façon pour construire un index inversé, qui à son tour va améliorer la performance du système de recherche.

### III.2.5.2. Vectorisation «vectorization»

Une requête sera souvent constituée de plusieurs termes formant une expression. Ce qui nécessite de passer par une étape de vectorisation de ses termes en fonction de leurs fréquences et enfin construire un vecteur TF «Terme Frequency» contenant les termes de la requête ordonnés par ordre décroisent de fréquences afin de faciliter l'étape d'appariement avec l'index inversé. Cette mesure permet d'évaluer l'importance d'un terme contenu dans une requête, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans la requête. Il varie également en fonction de la fréquence du mot dans l'expression. Le choix des termes à comparer avec l'index inversé dépend des fréquences par exemple on peut prendre les cinq mots les plus fréquents du vecteur TF et appliquer la fonction d'appariement avec l'index inversé.

### III.2.5.3. Appariement «Matching»

Dans cette étape il s'agit de faire un appariement entre le vecteur TF correspondant à la requête de l'utilisateur et l'index inversé du coran pour répondre d'une façon pertinente à un besoin spécifié.
Cette fonction peut être réalisée selon trois méthodes :

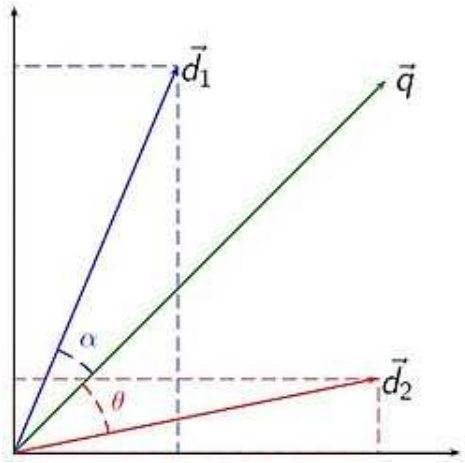
#### Modèle Booléen

Les documents sont représentés par des ensembles de termes et les requêtes traitées comme des expressions logiques. Considérant un vocabulaire $T = t_1, \ldots, t_m$, un document est caractérisé par la présence ou l'absence de chaque $t_i$ dans son contenu. La requête s'exprime alors avec des opérateurs logiques selon le formalisme de l'algèbre de Boole. Un document du corpus est ainsi considéré comme pertinent uniquement quand son contenu est vrai pour l'expression de la requête.

#### Modèle Vectoriel «cosine»

Il s'agit de trouver les documents qui répondent le mieux à une requête; qui est considérée comme un document, traduite en vecteur, et comparée aux vecteurs contenus dans le corpus des documents indexés.

Étant donnée une représentation vectorielle d'un corpus de documents, on peut introduire une notion d'espace vectoriel sur l'espace des documents en langage naturel. On en arrive à la notion mathématique de proximité entre documents.




Représentation de deux document « $d_1$ et $d_2$ » et d'une requête « $q$ » dans un espace vectoriel. La proximité de la requête aux documents est représentée par les angles $\alpha$ et $\theta$ entre les vecteurs.

En introduisant des mesures de similarité adaptées, on peut quantifier la proximité sémantique entre différents documents. Les mesures de similarité sont choisies en fonction de l'application. Une mesure très utilisée est la similarité cosinus, qui consiste à quantifier la similarité entre deux documents en calculant le cosinus entre leurs vecteurs. La proximité d'une requête $q$ à un document $d_1$ sera ainsi donnée par:

$$\cos\alpha = \frac{\mathbf{d_1} \cdot \mathbf{q}}{\|\mathbf{d_1}\| \, \|\mathbf{q}\|}$$

En conservant le cosinus, nous exprimons bien une similarité. En particulier, une valeur nulle indique que la requête est strictement orthogonale au document. Physiquement, cela traduit l'absence de mots en commun entre $q$ et $d_1$. De plus, cette mesure n'est pas sensible à la norme des vecteurs, donc ne tient pas compte de la longueur des documents.

Un avantage de la similarité cosinus est qu'elle peut efficacement profiter d'une implémentation par index inversé à condition d'indexer également la norme des documents. Chaque élément non nul de la requête $q$ permet de retrouver des documents potentiellement pertinents et le produit scalaire « numérateur de la similarité cosinus » est simultanément calculé par accumulation « en ligne ».

### III.2.6. Organigramme de recherche ‹Auto›

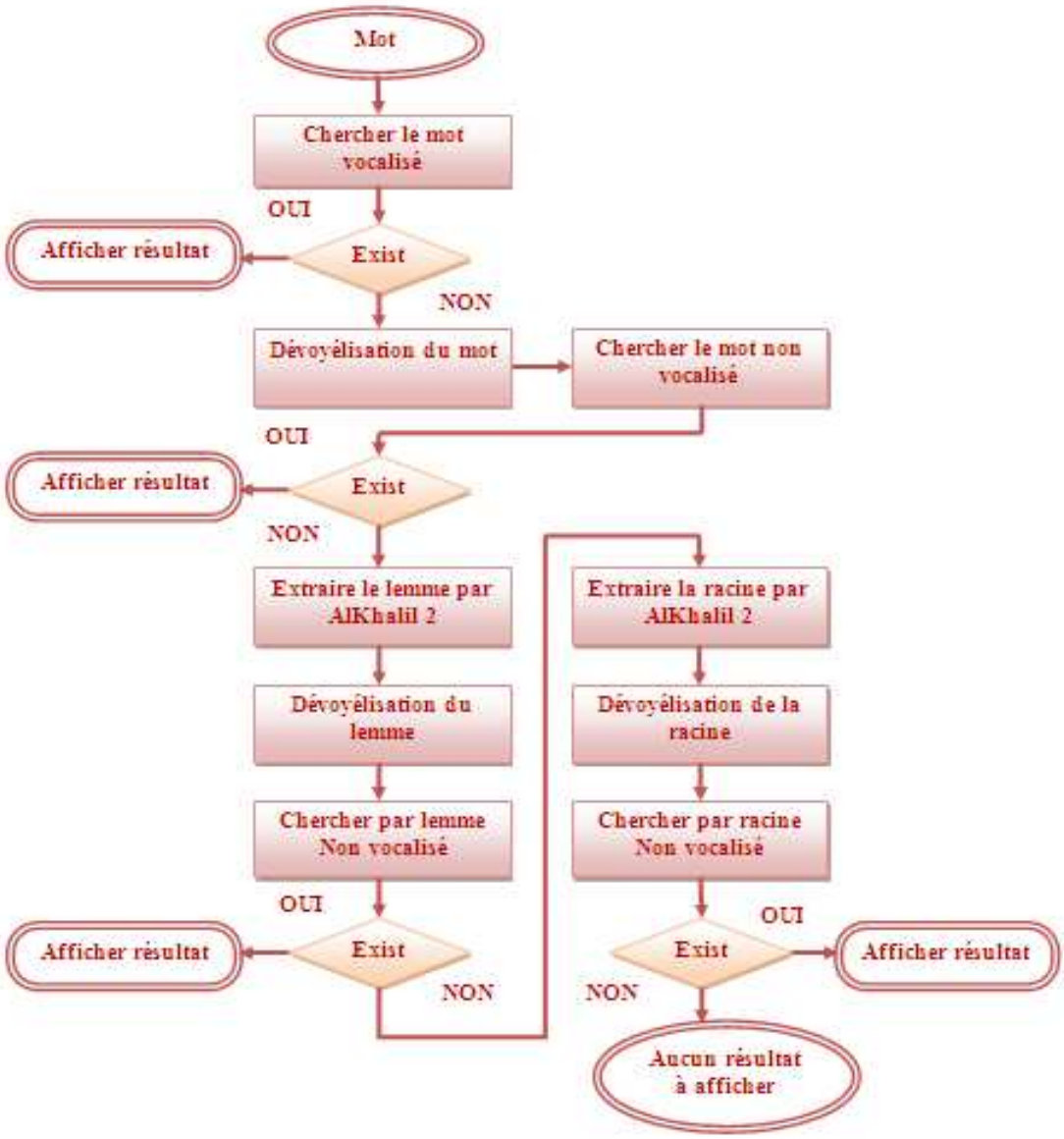



**Figure 14.** Organigramme de recherche ‹Auto›.

### Principe du fonctionnement

Dans le mode de recherche automatique, la requête de l'utilisateur suit le chemin suivant :

Une recherche est lancée par la requête à la forme introduite, dans ce cas on est entrain d'encadrer quand même la possibilité que le mot soit vocalisé ce qui est presque impossible, car l'utilisateur ne peut jamais entrer un mot vocalisé exactement comme dans le coran.

Si le résultat est vide, alors on passe à une recherche par mot non vocalisé, et c'est le type le plus efficace car la majorité des utilisateurs s'expriment par des requêtes non vocalisées, une étape de dévoyélisation de requête est nécessaire avant de lancer la recherche, sinon afficher le résultat retourné.

Lorsque les deux types de recherche précédents ne donnent aucun résultat, alors le mode automatique passe à l'extraction des lemmes et racines du mot par AlKhalil, toujours en passant par une étape de dévoyélisation, et dans ce cas la possibilité de répondre pertinemment à l'utilisateur est limitée, mais si elle existe elle sera vraiment précise, sinon aucun résultat à retourner.

### III.2.7. Organigramme de recherche ‹Tout›


**Figure 15.** Organigramme de recherche ‹Tout›.

### Principe du fonctionnement

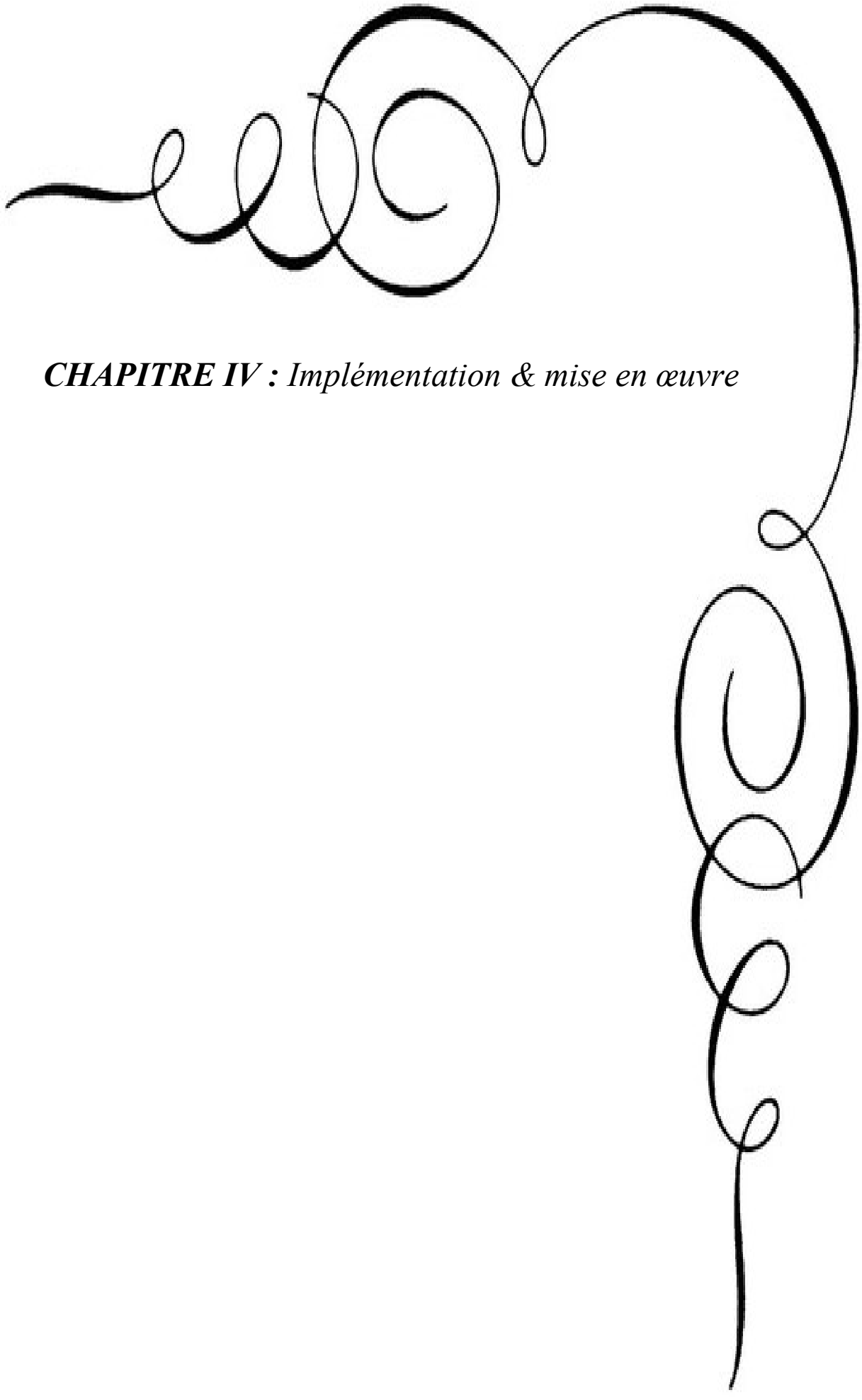
Dans ce mode de recherche, la requête de l'utilisateur suit le chemin suivant :

Une recherche est lancée par la requête à la forme introduite, pour la même raison que le mode automatique, ensuite, on passe à une recherche par mot non vocalisé, dans ce cas une étape de dévoyélisation de requête est nécessaire avant de lancer la recherche, puis, on passe à l'extraction des lemmes et racines du mot par AlKhalil, toujours en passant par une étape de dévoyélisation, et dans ce cas la possibilité de répondre à l'utilisateur augmente, avec plus de résultats.

## III.3. Conclusion

Ce chapitre a été consacré à la description conceptuelle de notre approche pour l'indexation, la recherche et le référencement coranique. La base relationnelle du Coran est présentée et le processus de référencement est expliqué. Deux schémas de recherche sont proposés pour répondre à n'importe quelle requête libre.

Apres la conception et la réalisation des différents diagrammes, on peut passer à l'implémentation de notre système d'indexation et d'exploration sémantique coranique dans les textes arabes, ce qui sera décrit dans le chapitre suivant.



***CHAPITRE IV :*** *Implémentation & mise en œuvre*

## IV.1. Introduction

Nous présentons dans ce chapitre les outils utilisés pour le développement de notre système, et son mise en pratique. Nous validons notre travail par la description de différentes expérimentations réalisées pour le référencement et la recherche.

## IV.2. Ressources utilisé

Les ressources physiques exploitées :

- Processeur Intel(R) Corre(TM) i3-2350M CPU @ 2.30GHZ.
- Mémoire vive d'une capacité de 4Go.

Et comme ressource logicielle, nous avons utilisé :

- Système d'exploitation : Windows7.
- Langage de programmation : JAVA.
- L'EDI : NetBeans de version 7.4

Notre choix s'est porté sur cet EDI car il permet d'intégrer une interface graphique, en utilisant la syntaxe du langage JAVA. Il offre au programmeur un environnement intégré pour la programmation orientée objet, visuelle et événementielle.

### IV.2.1. Pourquoi Java ?

Java est un langage de programmation très utilisé, notamment par un grand nombre de développeurs professionnels, ce qui en fait un langage incontournable actuellement.
On a travaillé avec java car il a beaucoup de caractéristiques parmi lesquelles **:**

- Son excellente portabilité **:** une fois votre programme crée, il fonctionnera automatiquement sous Windows, Mac, Linux etc.
- on peut faire de nombreux types de programmes avec Java **:**
  - ✓ des applications sous forme de fenêtre ou de console ;
  - ✓ des applets, qui sont des programmes Java incorporé à des pages Web ;
  - ✓ des applications pour appareils mobiles, comme les Smartphones, avec Java ME ﴿Java Micro Edition﴾ ;
  - ✓ des sites Web dynamiques avec J2EE ﴿Java 2 Entreprise Edition﴾ ;
  - ✓ et bien d'autre **:** JMF ﴿Java Media Framework﴾, J3D pour la 3D...

### IV.2.2. Pourquoi NetBeans ?

NetBeans est un ﴿EDI﴾, open source et multi-langue, créé par Sun et racheté par Oracle. Il a la particularité d'être multiplateforme **:** il est compatible avec Windows, MacOs, Linux et Solaris. Il permet d'intégrer une interface graphique en utilisant la syntaxe du langage JAVA.

### IV.2.3. Pourquoi SQL ?

Le SQL, est un langage Standard permettant à un client de communiquer des instructions à la base de données. Il se décline en quatre parties :

- le DDL comporte les instructions qui permettent de définir la façon dont les donnéessont représentées.
- le DML permet d'écrire dans la base et donc de modifier les données.
- le DQL est la partie la plus complexe du SQL, elle permet de lire les données dans labase à l'aide de requêtes.
- le DCL, qui ne sera pas vu dans ce cours permet de gérer les droits d'accès aux données.

A cela s'ajoute des extensions procédurales du SQL (appelé PL/SQL en Oracle). Celui-ci permet d'écrire des scripts exécutés par le serveur de base de données.

### IV.2.4. Pourquoi MySQL ?

- Rapide : Le serveur MySQL est très rapide.
- Facile à utiliser

MySQL est beaucoup plus simple à utiliser que la plupart des serveurs de bases de données commerciaux.

- API diverses

On peut effectuer diverses opérations sur une base MySQL en utilisant des interfaces écrits en C, Perl, Java, Python, PHP.

- Connexion et Sécurité

MySQL dispose d'un systéme de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.

- Portabilité

MySQL tourne sur divers système tels qu'Unix, Windows, Linux ou OS/2.

- Distribution ouverte

Les sources étant fournies, il est possible d'améliorer MySQL.

### IV.2.5. Pourquoi UML ?

UML est un langage graphique de modélisation des données et des traitements. C'est une formalisation très aboutie et non-propriétaire de la modélisation objet utilisée en génie logiciel.

### IV.2.6. Pourquoi ArgoUML ?

C'est un logiciel libre de création de diagramme UML. Programmé en JAVA, il est édité sous licence BSD. Il est multilingue et supporte la génération de classes JAVA (et même en d'autre langage de programmation).

## IV.3. Présentation de l'application

Le but principal de notre application est de construire un système d'indexation et d'exploration sémantique coranique dans les textes arabes.

Pour cela nous avons réalisé cette application «**Quran Referencing الاحالة القرانية**» qui permet à l'utilisateur de **:**

1) Réaliser un référencement coranique **:**

- En chargeant un corpus et en sélectionnant un mot, un paragraphe ou même tout le texte.
- Selon trois modes de référencement «احالة خفيفة, احالة متوسطة,احالة عالية ».
- Par «الافعال , الاسماء,الاسماء و الافعال ».

2) Exprimer une requête pour chercher dans le texte **coranique :**

Mode Auto :

- Selon organigramme de recherche ﴿Auto﴾.

Mode Tout :

- Selon organigramme de recherche ﴿Tout﴾.
- Rechercher directement par des lemmes et des racines existants.

## 1. Principale


**Figure 16.** Interface principale.

## 2. Les fonctionnalités de l'application

a. **Référencement coranique :** en cliquant sur l'onglet «الإحالة القرآنية»

Cet onglet nous permet de réaliser un référencement coranique dans les textes arabes, en sélectionnant un paragraphe ou même tous le texte.

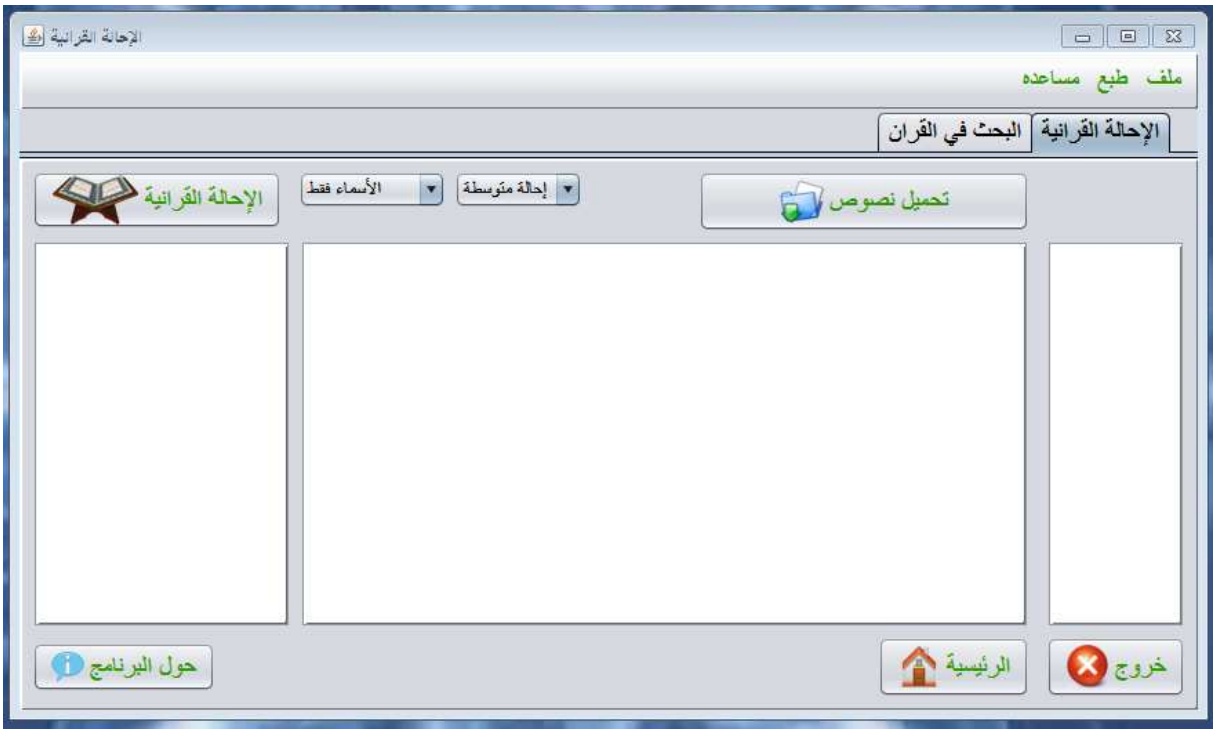



**Figure 17.** Référencement coranique.

On commence par le téléchargement du corpus en cliquant sur «تحميل نصوص».

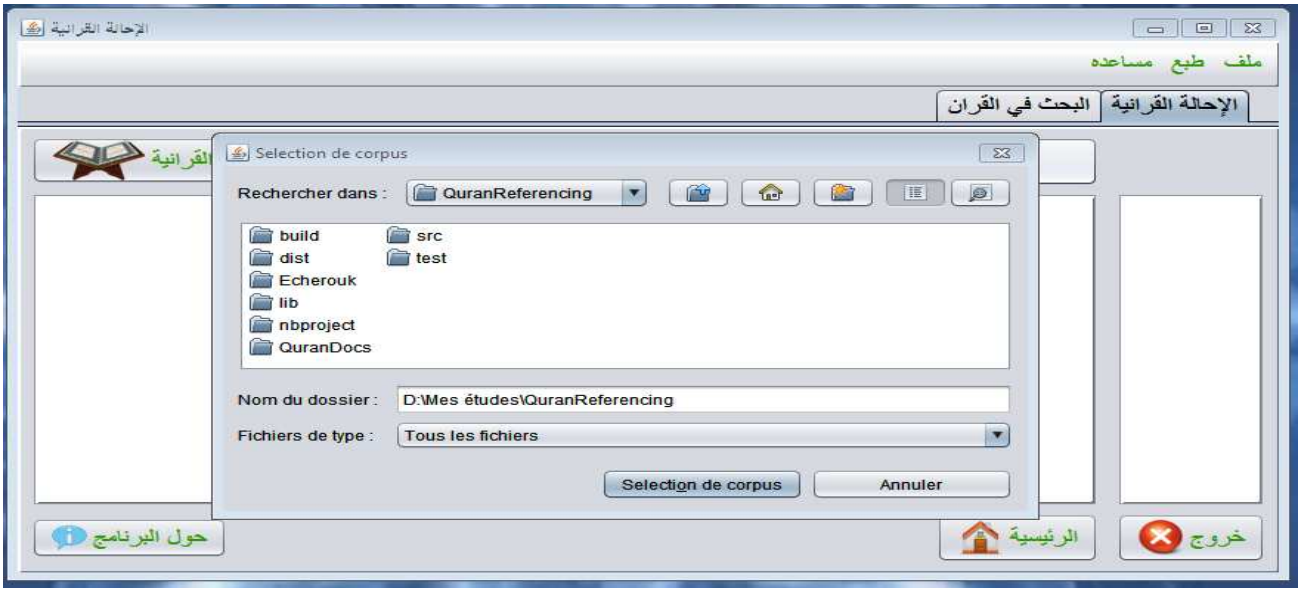



**Figure 18.** Téléchargement du corpus.

Choisir le mode du référencement «إحالة خفيفة, إحالة متوسطة,إحالة عالية » puis le type du mot «الأفعال , الأسماء,الأسماء والأفعال» et sélectionner un paragraphe à référencer ﴿on peut sélectionner tous le texte﴾.

Le niveau de référencement varie entre, léger, moyen et fort ; cette option est traduite par deux mécanismes lors du processus de filtrage et de mise en requête SQL.

Le filtrage désigne la fonction de sélection des lemmes descripteurs du texte arabe selon leur poids (TF). Dans un référencement léger, le seuil de filtrage est fixé au minimum (TF=1), alors qu'il se voit élevé pour les deux autres et par conséquent on ne fait l'appariement que par les lemmes descripteur les plus fréquents.

La mise en requête SQL vise à sélectionner les versets correspondants aux lemmes descripteurs du texte. Le niveau de référencement intervient dans le nombre d'occurrences présentes dans le même verset. Dans un référencement léger se nombre et réduit à 1 mais il augmente dans les deux autres niveaux.

Prenons un exemple avec «إحالة خفيفة» et «الأسماء والأفعال», on commence par la sélection d'une partie du texte.


**Figure 19.** Exemple avec «إحالة خفيفة» et «الأسماء والأفعال».

Appliquer le référencement en cliquant sur le bouton «الإحالة القرآنية»


**Figure 20.** Appliquer le référencement du paragraphe sélectionné.

Afficher le résultat du référencement dans l'onglet «البحث في القران».

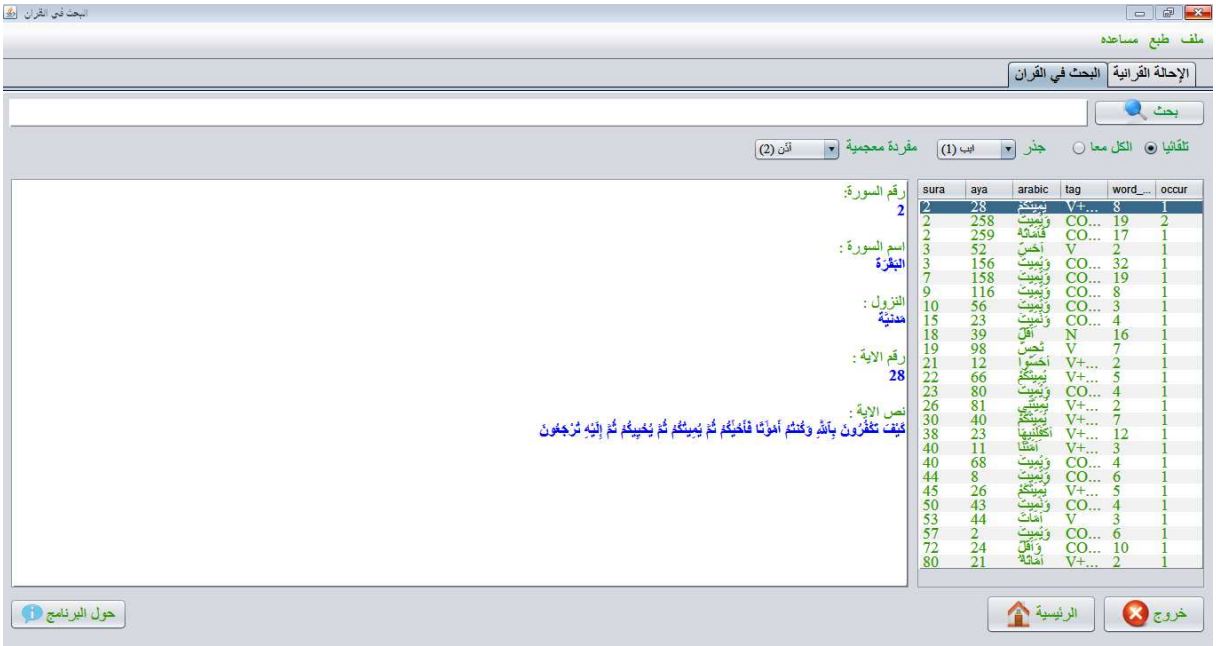



**Figure 21.** Affichage des versets référençant le paragraphe sélectionné.

a. **Recherche coranique :** en cliquant sur l'onglet «البحث في القران».


**Figure 22.** Recherche coranique.

Avec la possibilité d'afficher des versets, en sélectionnant une ligne dans la table, après le lancement d'une recherche selon le choix du lemme ou racine proposé par le système. L'exemple suivant est l'affichage du $11^{éme}$ verset du $4^{éme}$ chapitre, correspondant à la racine «ابو».

Gérer le cas ou un utilisateur lance une recherche sans entrer un mot.


**Figure 23.** Signal pour entrer un mot avant de lancer une recherche.

Lancer une recherche selon le mode Auto «تلقائيا» du mot «مريم» et voici les résultats :


**Figure 24.** Nombre de résultats dans les différents versets «تلقائيا».

Affichage des résultats obtenu du mot «مريم» .

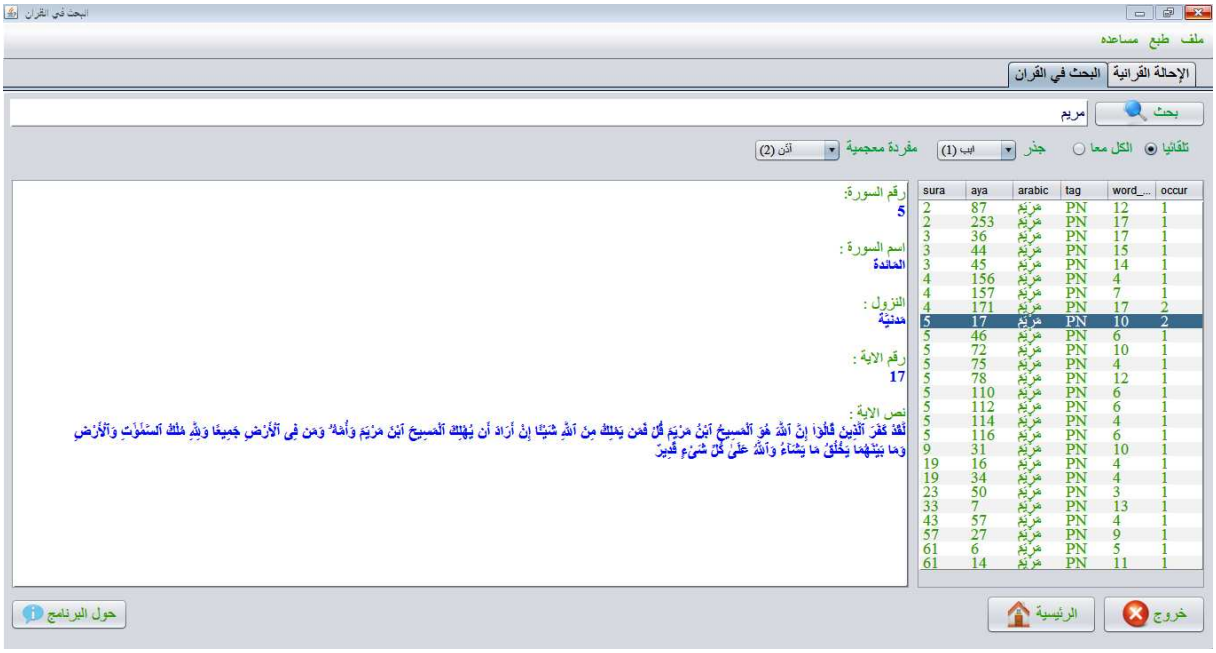



**Figure 25.** Affichage des résultats obtenu du mot «مريم» en mode Auto «تلقائيا».

Lancer une recherche selon le mode Tout «الكل معا» du mot «مريم» et voici les résultats :


**Figure 26.** Nombre de résultats dans les différents versets «الكل معا».

Les résultats d'une recherche selon le mode Tout «الكل معا» du mot «مريم» :

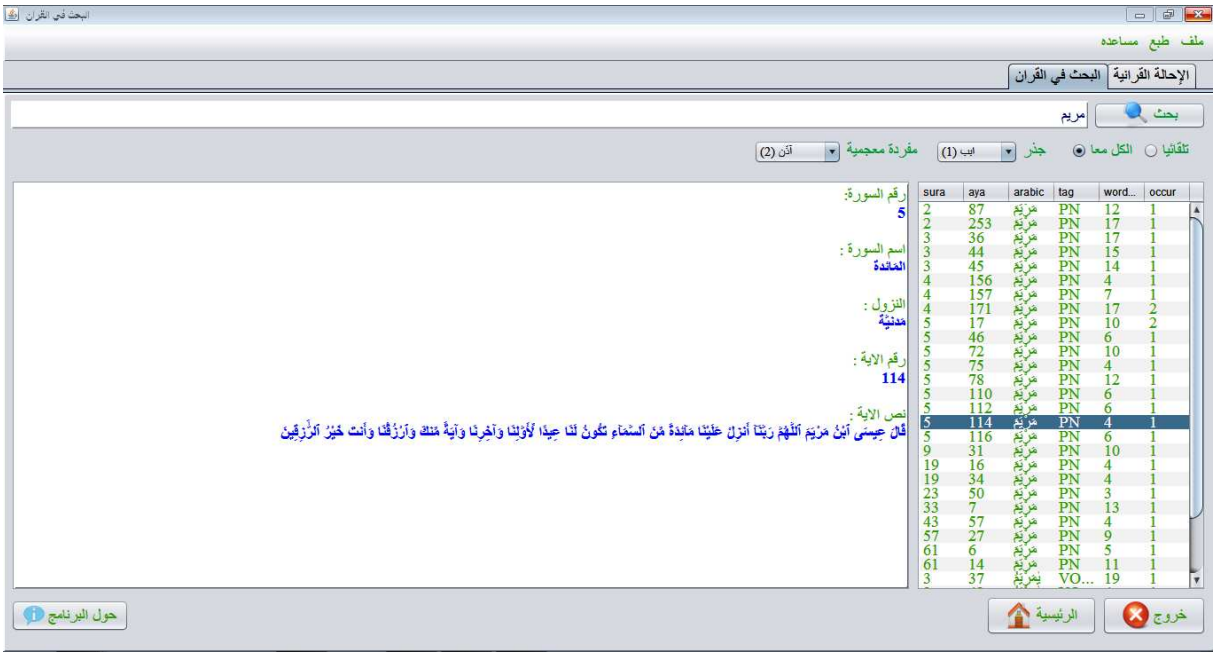



**Figure 27.** Affichage des résultats obtenu du mot «مريم» en mode «الكل معا».

Discussion des résultats de recherche pour le mot «مريم» avec les deux modes :

Auto : la recherche avec ce mode a retourné 28 résultats et c'est logique, car la fonction de recherche s'arrête des que le mot est trouvé au niveau de la base, et donc elle ne passe pas à une recherche par lemmes ou même par racines, ce qui ne permet pas de retourner tous les résultats possibles.

Tout : le nombre de résultat retourné dans ce mode est 35, en rajoutant des formes agglutinées. D'autres applications de recherche du Coran loupent cet aspect en déclarent un nombre inférieur pour le même mot recherché «مريم ». Dans ce cas les résultats retournés prouvent que la recherche par lemme et racine du mot ajoute une valeur, et augmente la perfection de la recherche, ce qui permet de retourner des résultats répondant le mieux au besoin des utilisateurs.

**Autres fonctionnalités supplémentaires :**

On peut lancer une recherche avec une expression comme pour un mot et voici un exemple en mode Auto «تلقائيا», avec «صلى وزكى».

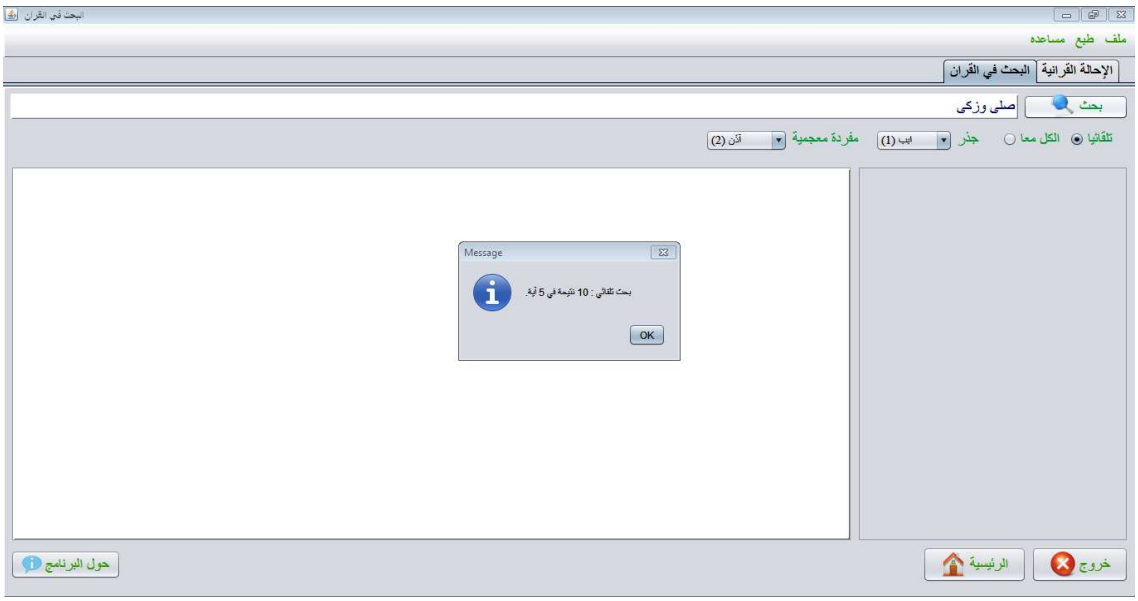



**Figure 28.** Nombre de résultats dans les différents versets «تلقائيا».

Affichage des résultats obtenu par l'expression «صلى وزكى» en mode Auto «تلقائيا».

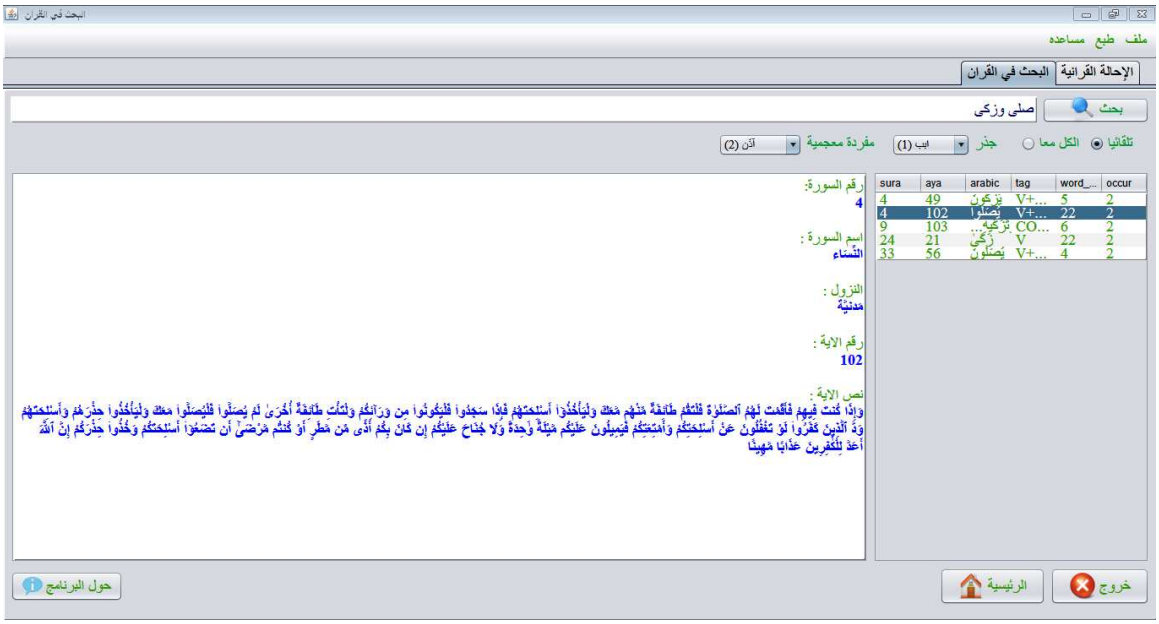



**Figure 29.** Affichage des résultats de l'expression «صلى وزكى».en mode Auto «تلقائيا».

Le même exemple en mode Tout «الكل معا».

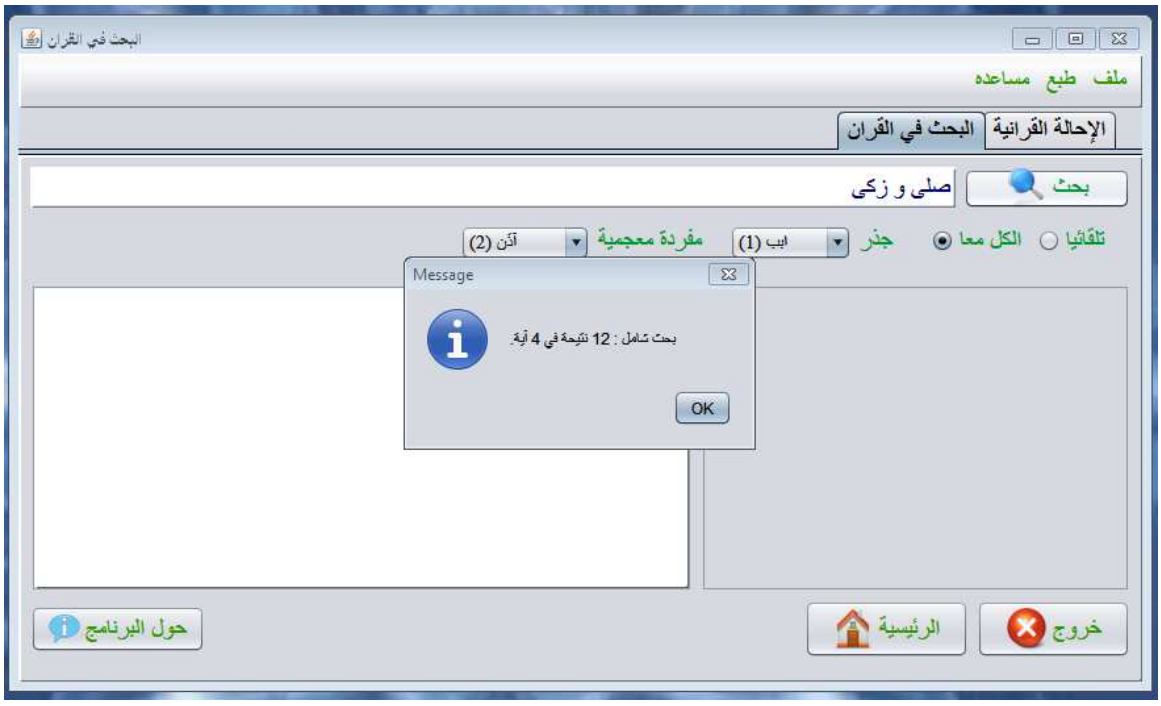



**Figure 30.** Affichage des résultats de l'expression «صلى وزكى».en mode Tout «الكل معا».

Affichage des résultats obtenus par l'expression «صلى وزكى».

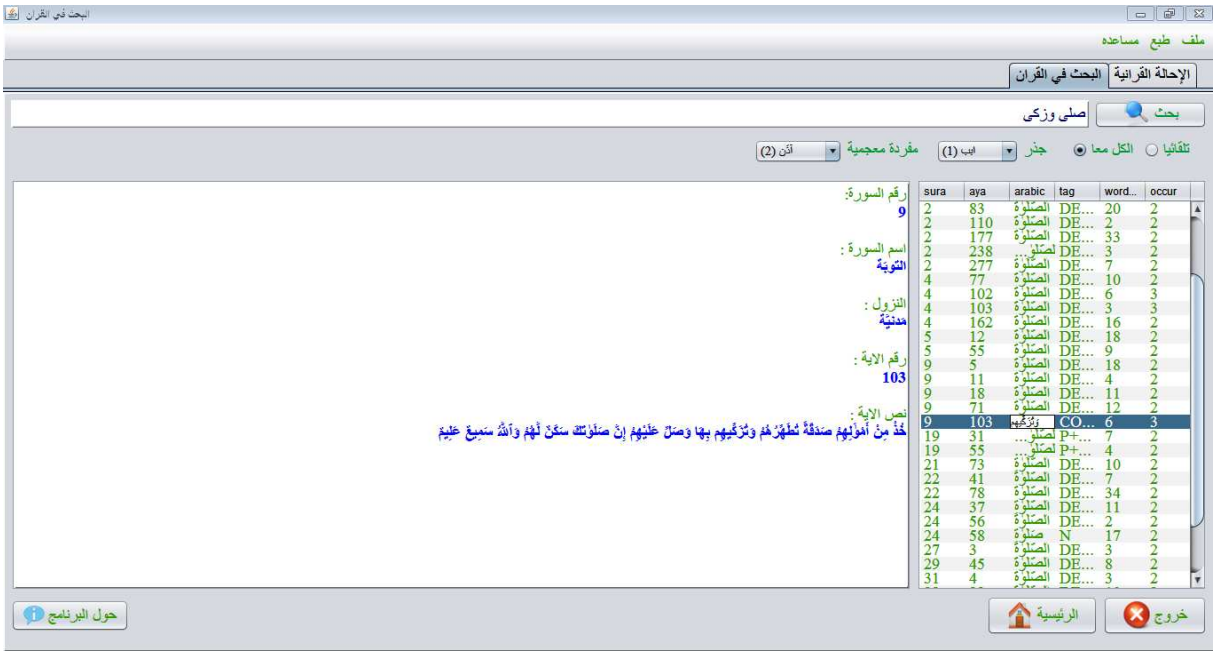



**Figure 31.** Affichage des résultats obtenu en mode Tout «الكل معا».

Cette application fournie d'autres fonctionnalités telles que l'impression, la copie, et l'aide.

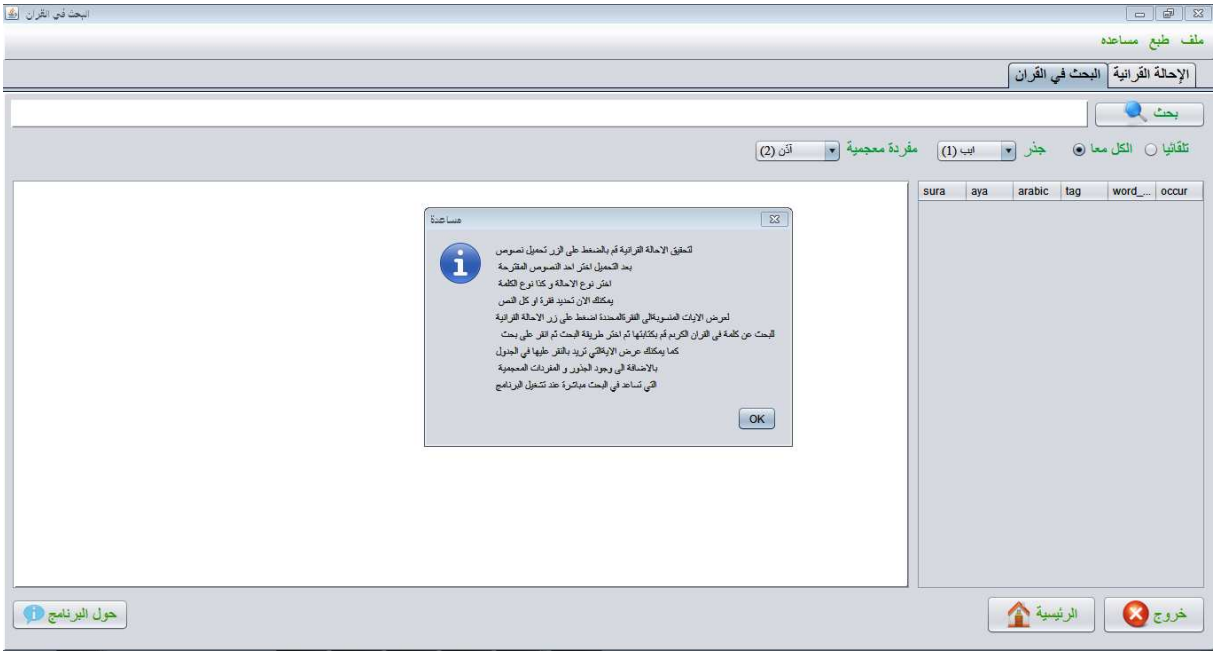



**Figure 32.** Affichage de l'aide.

## IV.4. Conclusion

Ce chapitre a été consacré aux aspects techniques d'implémentation et de mise en œuvre de notre système d'indexation et d'exploration coranique dans les textes arabes. Notre application consiste à réaliser un référencement coranique, et même faire une recherche coranique dans un temps raisonnable, tout en tenant conte de la qualité des résultats obtenus.

La structuration de la base Quran nous permet de remonter au lemme et à la racine du mot coranique pour en faire un appariement adéquat. Comme cette application est développée en JAVA et sous NetBeans, elle peut être déployée ultérieurement sous des systèmes mobiles.

# CONCLUSION GÉNÉRALE

Un système de recherche d'information en langue arabe doit prendre en considération ses caractéristiques morphologiques et proposer des outils et des techniques afin de permettre son traitement automatique. Le Coran se distingue par le fait qu'il reste éternellement une source incontestable pour la langue arabe et pour les différents écrits religieux. Lorsqu'un spécialiste, ou même un simple utilisateur, explore ou rédige un texte arabe, il souhaite souvent trouver une citation coranique pour :

- appuyer ses propos,
- vérifier ces jugements,
- chercher des correspondances terminologiques ou,
- chercher d'autres contextes pour le même texte.

Ces tâches relèvent certes de l'intelligence humaine, mais peuvent faire l'objet de plusieurs sujets d'étude et ouvrent de nouveaux perspectives pour l'exploitation des techniques de la RI et du text-mining au service de la promotion de la rédaction et de la lecture arabe.

Notre travail s'inscrit dans le cadre de l'indexation et l'exploration sémantique coranique dans les textes arabes. Nous avons étudié certaines caractéristiques de la langue arabe, notamment celles d'ordre morphologique. Contrairement aux autres langues, la langue arabe possède un système dérivationnel très riche et c'est dans cette caractéristique que réside la difficulté de son traitement. Ces caractéristiques constituent en effet les problèmes majeurs face aux travaux sur la langue arabe dans le domaine de la recherche d'information. Nous avons présenté les différents travaux réalisés pour son traitement automatique, tel qu'AlKhalil qu'on a utilisé dans notre projet.

Nous nous sommes basés dans notre implémentation d'un système d'indexation et de référencement coranique sur deux grands axes : l'indexation et la recherche. En effet, la phase d'indexation consiste à construire au préalable une structure d'accès aux documents qui facilitera la phase de la recherche. Celle-ci consiste à retrouver les documents les plus pertinents par rapport à une requête donnée; plus la phase d'indexation est sophistiquée, plus la phase de recherche est facile.

Notre contribution consiste en la réalisation d'une plate-forme d'indexation et de recherche dans le texte arabe. L'architecture proposée pour le stockage, l'indexation et la recherche des unités morphologique dans le Coran, nous a permis de concrétiser un modèle de référencement linguistique.

Les différentes pistes explorées pendant ce travail nous ont amenées à conclure que cette plate-forme représente une phase importante pour aboutir à l'indexation et le référencement sémantique des textes arabes, ouvrant ainsi une perspective intéressante pour les recherches dans l'analyse intelligente du texte arabe et en particulier le texte coranique.

# RÉFÉRENCES BBLIOGRAPHIQUES

**[1]** Salton, G., & McGill, M... Introduction to Modern Information Retrieval. McGrawHill, New York, 1983.

**[2]** Abderrezak Brahmi. Contribution à la Recherche Intelligente sur le Web: Indexation Sémantique des Textes Non-Structurés. Thèse de Doctorat, USTO-Oran, 2013.

**[3]** Abbassi Meftah. Recherche d'information. Mémoire du Master, l'université d'ourgla.

**[4]** Sebastiani, F. (2005).Text categorization. In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, pp. 109-129.

**[5]** Su, X., & Khoshgoftaar, T.M. (2009). A survey of collaborative filtering techniques, Adv. in Artif. Intell., Vol. 2009, pp. 1-19.

**[6]** Lloret, E., & Palomar, M. (2010). Challenging Issues of Automatic Summarization: Relevance Detection and Quality-based Evaluation. Informatica (Slovenia) Vol. 34(1): pp. 29-35.

**[7]** Efthimiadis E. (2000). Interactive query expansion: a user based evaluation in relevance feedback environment. Journal of the American Society for Information Science, Vol. 51, no 11, pp. 989-1003.

**[8]** Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. IBM Journal of Research and Development.

**[9]** GHOUL DHAOU. Outils génériques pour l'étiquetage morphosyntaxique de la langue arabe : segmentation et corpus d'entraînement. Dumas, version1, 12 Oct 2011.

**[10]** Fouad Soufiane Douzidia, Résumé automatique de texte arabe. Département d'informatique et de recherche opérationnelle, Faculté des arts et des sciences, Septembre, 2004.

**[11]** Lamia Hadrich Belguith, Chafik Aloulou & Abdelmajid Ben Hamadou, MASPAR : De la segmentation à l'analyse syntaxique de textes arabes. Laboratoire de Recherche LARIS – MIRACL, Faculté des Sciences Economiques et de Gestion de Sfax.

**[12**] Larkey, L. S., Ballesteros, L., & Connell, M. E. (2002). Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In Proceedings of SIGIR'2002, pp. 275-282, Tampere, Finland.

**[13]** Khoja, S., & Garside, R. (1999). Stemming Arabic text. Technical report, Computing Department, Lancaster University, Lancaster.

**[14]** Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium, University of Pennsylvania. LDC Catalog No.: LDC2002L49.

**[15]** Zargayouna H., « Quelle évaluation pour la Recherche d'Information Sémantique »,Troisième Atelier Recherche d'Information SEmantique RISE@CORIA'2011, 2011.

**[16]** http://www.nongnu.org/aramorph/ consulté le 20/01/2016 à 18:00H.

**[17]** Shereen Khoja (2001). ―APT: Arabic Part-of-speech Tagger. Proceedings of the Student Workshop at NAACL-2001, 2001.

**[18]** Shereen Khoja (2001). ―A tagset for the morphosyntactic tagging of Arabic. Corpus Linguistics 2001 conference, Lancaster.

**[19]** محمد زكي خضر و أكرم محمد زكي. دراسة احصائية لكلمات القرآن الكريم .المؤتمر الثالث للغة العربية وأدابها "الاتجاهات الحديثة في الدراسات اللغوية والأدبية. ص 287-302. الجامعة الإسلامية العالمية بماليزيا، 2011/9/30م

**[20]** http://comprendre-islam.com/le-decoupage-du-coran/ consulté le 06/05/2016 à 15:36H.

**[21]** http://www.easyquran.com/en/first.htm consulté le 18/04/2016 à 13:08H.

**[22]** http://tanzil.net/wiki/Tanzil_Project consulté le 15/02/2016 à 23:08H.

**[23]** http://corpus.quran.com/java/overview.jsp consulté le 05/03/2016 à 17:02H.

**[24]** http://quranytopics.appspot.com/ consulté le 18/01/2016 à 02:28H.

**[25]** http://www.qub.ac.uk/cite2write/harvard31.html consulté le 13/02/2016 à 19:35H.