République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Akli Mohand Oulhadj de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Mémoire de Master

en Informatique

Spécialité : *ingénierie des systèmes d'information de logiciel*

Thème

Open source Hadith Search Engine

Réalisé par :

- **Goura.Atika**
- **Remmak.Chahira**

Encadré par :

Mr TAHA ZERROUKI

2018/2019

REMERCIEMENTS

Tout d'abord, Je tiens à remercier Allah taala qui m'a donné la force et la patience d'accomplir ce modeste travail.

Je voudrais également remercier superviseur Dr. Taha Zerrouki*, qui a eu la gentillesse de mettre connaissances et expérience incomparables 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.

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.

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, Mes sœurs et Toute ma famille et Mes amies, et tous mes enseignants et mes collègues à l'université de Bouira, tous ceux que j'aime et tous ceux qui m'aiment, qu'ils trouvent ici l'expression de toute ma reconnaissance.

ATIKA.CHAHIRA

Résumé :

Al-hadith diffère de tous les documents que nous connus car il contient des connaissances dans tous les aspects de la vie. Sans lui, nous avions perdu la Sunna et l'héritage de la prophétie, il est donc très important de considérer le hadith et d'étudier toutes les informations qui s'y rapportent, et de savoir si le hadith est vrai ou non.

Et en raison du grand nombre de ces informations, il était nécessaire de trouver un moyen de les extraire. Il n'existe actuellement aucune méthode efficace, à l'exception de certains dictionnaires imprimés et de certains sites qui prennent en charge la recherche simple.

Nous avons donc développé d'un moteur de recherche open source dans al-hadith basé sur "elasticsearch", qui permet d'effectuer une recherche texte intégral et de trier les documents par pertinence en temps réel pour nous fournir une interface riche de fonctionnalités. Et il devrait être basé sur des méthodes modernes de recherche d'informations afin d'obtenir une bonne stabilité et une grande vitesse de recherche. Ce travail est utile pour les chercheurs et les spécialistes d'al-hadith.

Mots-clés: al-hadith, moteurs de recherche, elasticsearch.

Abstract :

The hadith differs from all the documents we know because it contains knowledge in all aspects of life. Without it, we lost the Sunnah and inherited the prophecy, so it is very important to study the hadith, study all the information related to it, and know whether the hadith is true or not.

Due to the large number of this information, there was a need to find a way to extract it. There is currently no effective method except for some printed dictionaries and some sites that support simple serial search in standard terms.

So we discussed the development of an open source search engine in speech based on the "elasticsearch", which allows full-text search and sorting of documents by relevance in real time to provide us with a full interface of features. It should be based on modern methods of information retrieval in order to obtain good stability and a high speed of research. This work is useful for researchers and hadith scholars.

Keywords: the hadith, search engines, elasticsearch.

الملخص :

الحديث يختلف عن جميع الوثائق التي نعرفها فهو يحوي المعارف في جميع جوانب الحياة.وبدونه كنا قد فقدنا السنة و وراثة النبوة , لذلك من المهم للغاية النظر في الحديث ودراسة جميع المعلومات المتعلقة به , ومعرفة ما إذا كان الحديث صحيح ام لا.

و لكثرة هذه المعلومات جاءت الحاجة إلى إيجاد طريقة لاستخراجها . و لا توجد وسيلة حالية فعالة باستثناء بعض المعاجم المطبوعة و بعض المواقع التي تعتمد البحث البسيط التسلسلي بالعبارات النمطية.

لذلك تطرقنا إلى تطوير محرك بحث في الحديث يعتمد على تطبيق "البحث المرن " الذي هو محرك بحث مفتوح المصدر, يسمح بإجراء بحث عن النص الكامل و فرز المستندات حسب الصلة في الوقت الفعلي ليقدم لنا واجهة مليئة بالميزات. و ينبغي الاستناد إلى الأساليب الحديثة في استرجاع المعلومات من أجل تحصيل استقرار جيد وسرعة عالية في البحث هذا العمل مفيد للباحثين ودارسي الحديث.

الكلمات المفتاحية : الحديث، محركات البحث , البحث المرن.

Table des matiéres

REMERCIEMENTS

Dédicaces

Résumé

Abstract

الملخص

Table des matières i

Liste des figures.....vi

Liste des tableaux .....…. viii

Liste des abréviations ix

Introduction général.....1

I. Etats de l’art

Chapitre 1 : Les aspects informationnels des hadiths

1. Introduction.....3

2. Définition 3

3. Structure d'al-hadith 3

3.1. Selon la chaine de narrateurs (Snad السند).....4

3.2. Matn (متن) 4

4. Classification d'al-hadith 4

4.1. Collection d’al-hadith (source مصدر الحديث).....4

4.2. Books d'al-hadith (كتب الحديث) 5
4.3. Chapter (باب الحديث) 5
4.4. Narrator (راوي الحديث) 5
4.5. Fiabilité d'al-hadith 6
4.6. Author (مخرج الحديث) 6
5. Aperçu de langue arabe 6
6. Outils de recherche de hadith 7
6.1. Al-Durar Al-Sunniya (موقع الدرر السنية) 7
6.2. Islam Web (موقع إسلام ويب) 8
7. Discussion 9
8. Conclusion 10

Chapitre 2: Les moteurs de recherche dédies

1. Introduction 11
2. Définition 11
2.1. Mot clé 11
2.2. Descripteur 11
2.3. Document 11
2.4. Requête 11
2.5. Pertinence 11
3. Moteurs de recherche 12
4. Recherche en texte intégral 12
5. Processus d'indexation 12
5.1. Définition et rôle de l'indexation 13
5.2. Les modèles d'indexation 13
5.2.1. L'indexation manuelle (indexation humaine) 13
5.2.2. L'indexation automatique 14
5.2.3. L'indexation semi-automatique 15
5.3. Type d'index 16
5.3.1. Index du document 16
5.3.2. Index à terme 16

5.3.3. Index inverse 17
5.4. Stockage d'index 18
5.5. Phases d'indexation 18
5.5.1. Tokénisation 18
5.5.2. Normalisation 19
5.5.3. Elimination des mots vides 19
5.5.4. Pondération 19
5.5.4.1. Pondération locale 19
5.5.4.2. Pondération globale 19
6. Requête 20
6.1. Notion de pertinence 20
6.2. Les modèles de recherche d'information 21
6.2.1. Modèle booléen 21
6.2.2. Modèle vectoriel 22
6.3. Processus de recherche 22
7. Recherche sémantique 23
8. Conclusion 24

Chapitre 3: Les fonctionnalités Elasticsearch

1. Introduction 25
2. Qu'est-ce qu'Elasticsearch ? 25
3. Anantages 25
4. Inconvénients 25
5. Fonctionnements d'Elasticsearch 26
5.1. Présentation d'Apache Lucene 26
5.2. Architecture générale 26
5.3. Format fondamentaux 27
5.3.1. JSON 27
5.3.2. API Rest 27
5.4. Rôle de Logstash et Kibana 28
5.4.1. Logstash 28
5.4.2. Kibana 28
6. Indexation d'un document 28
6.1. Recherche un document 29

6.1.1. Recherche simple 29
6.1.2. Recherche avec une Query-DSL 29
6.1.2.1. Requête 29
6.1.2.2. Filters 30
6.1.3. Agrégation 31
7. Conclusion 31

II.Conception et réalisation

Chapitre 4: Conception

1. Introduction 32
2. Difficultés de recherche dans l'al-hadith 32
3. Classification des fonctionnalités de la recherche d'al-hadith 33
3.1. Fonctionnalités de recherche avancée 34
3.2. Fonctionnalités supplémentaires 35
3.3. Aspect linguistiques 35
3.4. Fonctions de recherche dans al-hadith 36
4. Modélisation 36
4.1. Diagramme de classe (UML) 36
5. Méthode de recherche 41
6. Architecture de notre system 42
6.1. Traitement d'al-hadith 43
6.1.1. Prétraitement 44
6.1.2. Index inversé 44
6.1.3. Exemple de traitement de document (texte Al-hadith) 45
6.2. Traitement des requêtes 46
6.2.1. Exemple de requête 46
7. Conclusion 47

Chapitre 5: Implementation

1. Introduction 48
2. Ressources utilisé 48
2.1. Python 48

2.2. Elasticsearch 48
3. Pourquoi Open Source? 48
4. Implémentation des fonctionnalités de recherche 49
4.1. Présentation de l'application 50
4.1.1. Interface principale 50
4.1.2. Rechercher par terme 51
4.1.3. Rechercher par phrase 51
4.1.4. Rechercher par des options d'al-hadith structurée 52
5. Analyser et restaurer de données avec Kibana 54
6. Traitement d'un exemple des requêtes 56
7. Conclusion 57
Conclusion générale 58
Bibliographie 60

Table des figures

Figure 1.1 : Exemple d'AL-Hadith du livre de la foi (كتاب الايمان) dans Sahih Bokhari 3

Figure 1.2 : Construction d'un mot arabe complexe en utilisant la racine 7

Figure 1.3 : Aperçu du site Web Al-Durar Al-Sunniya 8

Figure 1.4 : Aperçu du site Web islam web 9

Figure 2.1 : Les modèles d'indexation..... 13

Figure 2.2 : Les points clés d'indexation manuelle 14

Figure 2.3 : Les méthodes de L'analyse de système d'indexation automatique..... 15

Figure 2.4 : Les étapes de L'indexation automatique..... 15

Figure 2.5 : Les phases de processus d'indexation 18

Figure 2.6 : Processus de recherche..... 23

Figure 3.1 : Les requêtes les plus importantes dans la recherche en texte intégral 30

Figure 3.2 : Les requêtes les plus importantes dans la recherche basée sur des termes 30

Figure 3.3 : Les types d'agrégation 31

Figure 4.1 : Classification des fonctionnalités de recherche d'al-hadith. 34

Figure 4.2 : Diagramme de classe d'al-hadith 37

Figure 4.3 : Prototype de base de recherche. 41

Figure 4.4 : L'architecture de notre système 42

Figure 4.5 : Traitement de texte AL-Hadith 43

Figure 4.6 : Structure d’index inversé..... 44

Figure 4.7 : Exemple d’al-hadith. 45

Figure 5.1 : Interface principale. 50

Figure 5.2 : Rechercher par terme 51

Figure 5.3 : Rechercher par sanad 51

Figure 5.4 : Rechercher par matn 52

Figure 5.5 : Rechercher par collection 52

Figure 5.6 : Rechercher par book 53

Figure 5.7 : Rechercher par chapter 53

Figure 5.8 : Recherche par narrator 54

Figure 5.9 : Le fond de discover tous les documents d'index al-hadiths 54

Figure 5.10 : Les 5 premier narrateurs ont raconté la plupart des hadiths 55

Figure 5.11: Écran de visualisation des genres classifié d'al-hadith 55

Liste des tableaux

Table 1.1 : les collections d'al-hadith plus importants. 5

Table 1.2 : Analyse des sites de recherche pour al-hadith. 9

Table 4.1 : Description des tables de la base Hadith 39

Table 5.1 : Implémentation des fonctionnalités de recherche 49

Table 5.2 : Index termes des hadiths pertinents pour la requête. 56

Table 5.3 : Score de hadiths. 57

Liste des abréviations

API : Application Programme Interface

IDF : Inverse Document Fréquence

TF : Terme Fréquence

TF*IDF : Terme fréquences - Inverse document fréquence

SRI : Système de recherche d'information

URL : localisateur uniforme de ressource

RI : Recherche d'information

RSV : Valeur d'état de récupération

XML : Extensible Mark-up Language

SGBDR : système de gestion de base de données relationnel

NoSQL : Not only SQL (Pas seulement SQL)

JSON : JavaScript Object Notation

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

SQL : Structured Query Language

SVM : les séparateurs à vaste marge

Introduction générale

Al-hadith est une source importante car il contient beaucoup d'informations dans tous les aspects de la vie, donc on ne peut extraire rien sauf manuellement, et cela ne suffit pas pour son importance dans nos vies. Sans lui, la Sunna et l'héritage de la prophétie auraient été perdus.

En raison de la grande quantité d'informations stockées dans al-hadith, il est devenu extrêmement difficile pour les moteurs de recherche habituels d'extraire avec succès les informations de base.

Il existe de nombreuses applications pour les besoins de recherche, et la plupart des applications avaient la fonction de recherche, mais simplement: une recherche simple.

Notre suggestion est de développer un moteur de recherche open source basé sur "elasticsearch" qui permet la recherche texte intégral et le tri des documents par pertinence en temps réel, pour fournir aux utilisateurs une interface riche en fonctionnalités et des développeurs d'API riches. Mais pour atteindre cet objectif, vous devez d'abord répertorier et classer toutes les fonctions de recherche possibles et utiles.

Nous avons organisé le rapport comme suit:

Chapitre 01: Les aspects informationnelle des hadiths.

Dans ce chapitre, nous expliquerons d'abord la définition d'al-hadith et la structure de ce dernier. Ensuite, nous discutons de la classification d'al-hadiths, Après cela étudier quelque Outils de recherche d'al-hadith.

Chapitre 02: les moteurs de recherche dédient.

Dans ce chapitre, nous expliquerons le fonctionnement des moteurs de recherche en expliquant leurs principales composantes: indexation et requête en définissant les concepts de base dans le domaine des systèmes de recherche d'informations puis nous expliquerons le processus de recherche et la notion de pertinence.

Chapitre 03: les fonctionnalités Elasticsearch.

Dans ce chapitre, nous expliquerons le fonctionnement d'Elasticsearch en expliquant ses principaux composants

Chapitre 04: Conception.

Ce chapitre présente les différentes architectures et schémas liés à la conception et à l'indexation des systèmes de recherche, puis nous suggérons plusieurs améliorations pour implémenter toutes les fonctionnalités de recherche possibles.

Chapitre 05: Implimentation :

Dans ce chapitre, nous présentons les outils utilisés pour développer et implémenter notre système, ainsi que quelques-unes de ses interfaces, et nous expliquons pourquoi nous avons décidé d'utiliser open source.

Etat de l'art

Chapitre 01 :

Les aspects informationnels des hadiths

1. Introduction :

Al-hadith est la deuxième source des règles de l'islam. Sans lui, nous aurions perdu sunna (السنة), par conséquent, il est important de considérer al-hadith, d'étudier toutes les informations qui s'y rapportent et de s'en occuper pour savoir si al-hadith est vrai ou non. la recherche sur al-hadiths préserve la Sunna (السنة) et l'héritage de la prophétie (النبوة). Par conséquent, nous devons développer un moteur de recherche dans al-hadith open source.

Notre sujet est l'accès à un tel système pour nous aider à trouver facilement al-hadith et donner toutes les informations contenues dans ce. Dans ce chapitre, nous expliquerons d'abord la définition d'al-hadith et la structure de ce dernier. Ensuite, nous discutons de la classification d'al-hadiths, Après cela étudier quelque Outils de recherche d'al-hadith.

2. Définition :

Al-hadith est dérivé du mot arabe "الحديث" qui signifie la nouvelle et l'histoire. Selon la branche sunnite de l'islam, l'al-hadith est un discours, une discussion, une action, une approbation ou une description physique ou morale attribuée au prophète Mohamed (محمد ﷺ). [Batyrzhan2014]

3. Structure d'al-hadith :

Pour examiner al-hadith, nous devons en connaître les composants: matn (le texte centrale المتن), sanad (la chaîne de narrateurs السند).[Azmi,Badia 2010]

الحديث = متن (نص الحديث) + سند (الإسناد وهو سلسلة الرجال الموصولة إلى المتن)

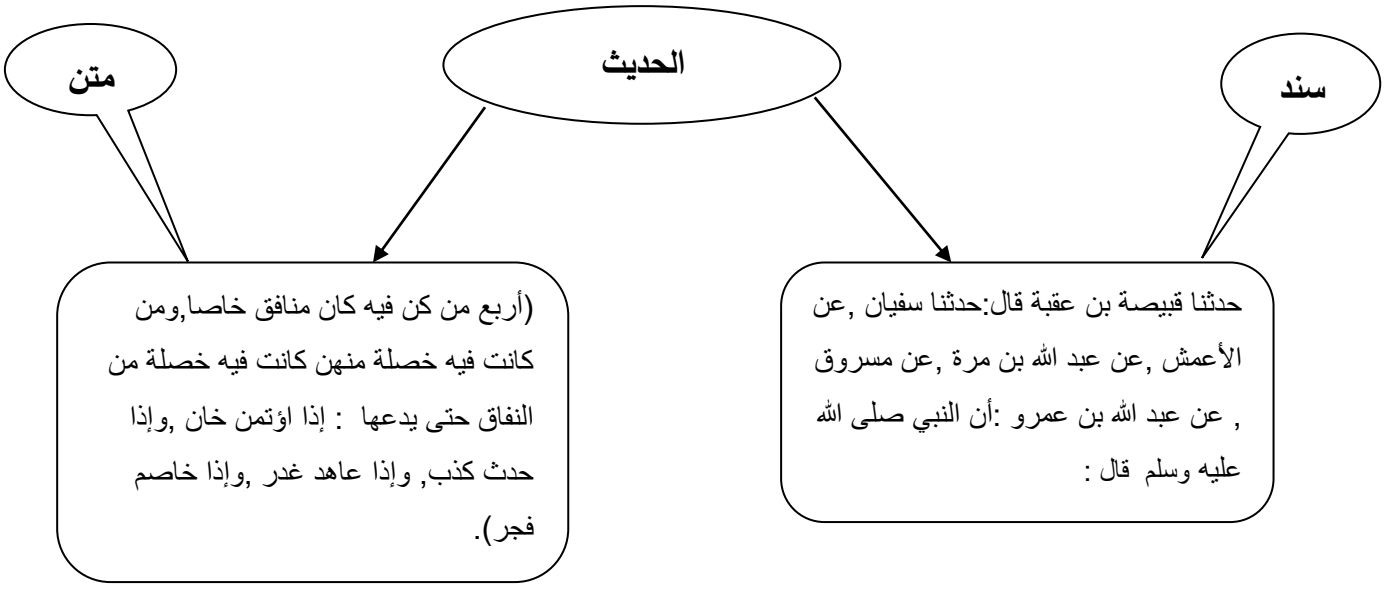


Figure 1.1 : Exemple d'al-hadith du book de la foi (كتاب الإيمان) dans Sahih Bokhari.

3.1. Chaine de narrateurs(Sanad السند) :

Sanad est la chaîne de narrateurs (الرجال) cela pointe vers le texte d'al-hadith. Sanad composé de tous ceux qui ont transmué le texte (al-hadith), en commençant par le dernier narrateur et se terminant par le Prophète (ﷺ محمد).

Pourquoi sommes-nous intéressés par l'attribution (sanad)? La réponse courte sera, cela aide à documenter d'AL-Hadith.

Sanad prend typique la forme suivante: Nom → Nom →→ •• → Nom → Le prophète Mahomed (ﷺ محمد) (fin de sanad).Dans le sanad, le mot "→" est généralement une phrase explicite, telle que "أخبرنا" ou "حدثنا". [الخطيب2010]

Voici un exemple d'al-hadith de Sahih al-Bokhari - book du début de la révélation (كتاب بدء الوحي).

حدثنا الحميدي عبد الله بن الزبير ,قال حدثنا سفيان, قال حدثنا يحي بن سعيد الأنصاري, قال أخبرني محمد بن إبراهيم التيمي ,أنه سمع علقمة بن أبي وقاص الليثي يقول, سمعت عمر بن الخطاب رضي الله عنه على المنبر قال سمعت رسول الله ﷺ يقول : إنما الأعمال بالنيات و إنما لكل امرئ ما نوى فمن كانت هجرته إلى دنيا يصيبها أو إلى امرأة ينكحها فهجرته إلى ما هاجر إليه. (صحيح البخاري – كتاب بدء الوحي – باب كيف كان بدء الوحي)

Sanad contient une chaine écrite de six narrateurs :

760م/143هـ) يحي بن سعيد الأنصاري → (توفي815م/198هـ) سفيان → (توفي834م/219هـ) الحميدي عبد الله بن الزبير عمر بن الخطاب→ (توفي705م/861هـ) علقمة بن أبي وقاص الليثي→ (توفي738م/120هـ) محمد بن إبراهيم التيمي→ (توفي (توفي644م/23هـ).

3.2. Matn :

Matn est le texte d'al-hadith qui vient après le lien de dire (السند) , de faire ou de rapporter. [الخطيب2010]

4. Classification d'al-hadith :

Afin de classer l'al-hadith, nous expliquons les méthodes de classification.

4.1. Collection d'al-hadith (source مصدر الحديث) :

Cette classification est basée sur des collections d'al-hadith célèbres. Chaque collection porte le nom du réalisateur (author مخرج الحديث) et constitue une collection de books. [عزمي2017]

Les plus importants collections d'al-hadith : [عزمي2017]

Collection (المجموعة)	Volume (عدد الأحاديث)
صحيح بخاري	7397
صحيح مسلم	12000
سنن ابن ماجه	4341
سنن أبي داوود	5274
سنن الترمذي	3956
سنن النسائي	5758
موطأ الإمام مالك	1942
مصنف ابن أبي شبيه	37251
مسند ابن حنبل	27647
سنن الدارمي	3455
صحيح ابن خزيمة	3079

Table 1.1 : les collections d'al-hadith plus importants.

4.2. Books d'al-hadith (كتب الحديث) :

La méthode de cette classification est de combiner al-hadiths d'un même sujet dans même collection, sous un titre commun, par exemple le livre de prière (كتاب الصلاة), le livre de Zakat (كتاب الزكاة, le livre de vente (كتاب البيوع). [نورالدين1997]

4.3. Chapter d'al-hadith (باب الحديث) :

Chaque book comprend un chapter ou un groupe de chapters et chaque chaptre comprend hadith ou plusieurs,et ce chapter est placé dans un titre indiquant le sujet, par exemple باب حلاوة الإيمان. [نورالدين1997]

4.4. Narrator d'al-hadith (راوي الحديث) :

Les six la plupart des narrateurs (الرواة) sont :

1 – أبو هريرة – رضي الله عنه – روى 5374 حديثًا.

2 – عبد الله بن عمر – رضي الله عنهما – روى 2630 حديثًا.

3 – أنس بن مالك – رضي الله عنه – روى 2286 حديثًا.

4 – عائشة أم المؤمنين – رضي الله عنها – روت 2210 من الأحاديث.

5 – عبد الله بن عباس – رضي الله عنهما – روى 1660 حديثًا.

6 – عبد الله بن جابر – رضي الله عنهما – روى 1540 حديثًا. [ذياب1434]

4.5. Fiabilité d'al-Hadith (حكم الحديث):

Il existe une approche organisée dans les sciences islamiques qui nous aide à faire la distinction entre al-hadith correct et al-hadith incorrect appelé "la science d'al-hadith" .Au fil du temps, de plus en plus de narrateurs ont été associés à l'attribution, raison pour laquelle l'attitude nécessitait des règles et des critères d'acceptation d'al-hadith bien définis: ces règles et normes sont connues sous la science d'al-hadith (علم الحديث). Le jugement final dans al-hadith déterminera sa catégorie:

- **Vrai hadith** (حديث صحيح): c'est un hadith dont les narrateurs sont crédibles et connus par leur bonne mémorisation et dont la chaine des narrateurs est continue.
- **Bien** qui signifie "bon" (حديث حسن) décrit récemment la validité d'al-hadith n'est pas confirmée comme vrai, mais acceptable : c'est un hadith dont les narrateurs sont crédibles, dont la chaine des narrateurs est continue. Mais dont la capacité de mémorisation des narrateurs est l'égarement faible par rapport au niveau requis. [طحان1985]
- **Faible** (حديث ضعيف) à cause :
  - d'une coupure dans la chaine de narrateurs, sachant qu'une coupure se traduit par l'absence d'un ou plusieurs narrateurs.
  - en raison d'un défaut imputé à l'un de ses narrateurs. [طحان1985]

4.6. Author (مخرج الحديث) :

Il est appelé celui qui a produit al-hadith à partir des imams et des érudits du hadith tels que: البخاري و مسلم. [نورالدين1997]

5. Aperçu de la langue arabe :

L'arabe est la langue maternelle de plus de 300 millions de personnes. Contrairement aux alphabets latins, l'écriture arabe est dirigée de droite à gauche. L'alphabet arabe est composé de 28 lettres. Les mots arabes ont deux types, féminin et masculin, trois nombres, singulier, double et pluriel, et trois cas grammaticaux: nominal (الاسمية), accusatoire(الفعلية) et héréditaire(شبه جملة) .

Le nom a un état nominal lorsqu'il est sujet, accusé lorsqu'il fait l'objet d'un acte et génitif lorsqu'il est l'objet de la préposition. Les mots sont classés en deux parties principales du discours, les noms (y compris les adjectifs et les circonstances), les verbes. [الحربي2008]

L'alphabet arabe comprend 25 consonnes et trois voyelles longues. En outre, le système alphabet utilise un total de treize signes différents pour représenter les voyelles courtes et est placé au-dessus

ou au-dessous de la lettre. Il n'y a pas de majuscule dans le texte arabe. Cela rend difficile l'identification des noms corrects. La morphologie arabe est très riche, mais systématique. Les mots arabes sont dérivés d'un ensemble de racines de consonnes (principalement une trilogie) contenant la signification de base du mot. Le fait de placer ces consonnes dans des motifs modifiera la signification de la racine pour créer une variété de mots liés, appelés stem. C'est ce qu'on appelle le processus de dérivation. De plus, les formes de texte arabes sont composées de préfixes d'empilement (par exemple, d'articles, de prépositions) et de suffixes (pronoms associés) sur les stems. La figure1.2 illustre le modèle de construction du mot arabe. [2017عزمي]

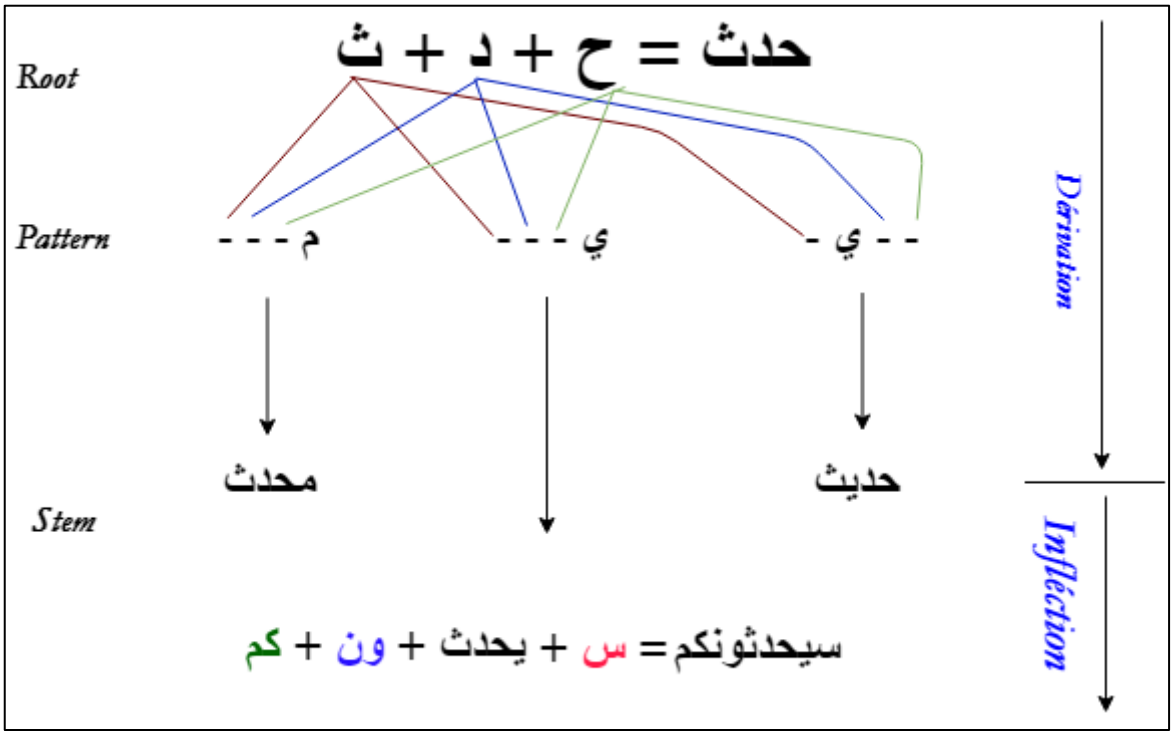


Figure 1.2: Construction d'un mot arabe complexe en utilisant la racine.

6. Outils de recherche d'al-Hadith :

Ici, nous avons listé quelques logiciels et sites web qui offrent la recherche dans al-hadith comme l'une de leurs principales options. Commençant par une petite description de chacun comprenant les fonctionnalités.

6.1. Al-Durar Al-Sunniya:(موقع الدرر السنية)

C'est en effet un site Web qui fournit des informations. La base de données de cesite démontre l'engagement des développeurs de sites Web à offrir la possibilité de faire participer les utilisateurs d'Internet et de faciliter la recherche d'al-hadith. Par conséquent, Durar Al-Sanniya est facile que la

recherche de texte, et comprend nombreuses Mawsuaats (موسوعات), telles que موسوعة العقيدة ,موسوعة الأديان .

Figure 1.3 : Aperçu du site Web Al-Durar Al-Sunniya

6.2. Islam Web : (موقع إسلام ويب)

C'est un site Web, il ne traite pas de connaissances scientifiques d'al-hadiths, mais le contenu du site Web couvre d'autres connaissances, telles que l'Aqidah (العقيدة), le Coran (القرأن), le Fiqh (الفقه), le Seera (السيرة), etc. En outre, Islam web est simple et complet, ce qui en fait un site précieux.

Figure 1.4 : Aperçu du site Web islamweb

7. Discussion :

Nous avons effectué une analyse des sites de recherche pour al-hadith, pour obtenir le meilleur site. Les résultats sont présentés dans le tableau suivant:

Les informations d'al-hadith	إسلام ويب	الدرر السنية
Sanad (السند)	√	X
Matn (المتن)	√	√
Narrator (الراوي)	X	√
author (التخريج)	X	X
Fiabilité(حكم)	√	√
Collection(مصدر)	√	√
Book (كتاب)	√	X
Chapter (الباب)	X	X
Position d'al-hadith	√	X
Open source	X	X
API	X	X

Table 1.2 : Analyse des sites de recherche pour al-hadith.

Le site (إسلام ويب) qui contient des informations six relatives au hadith, suivi du (الدرر السنية) avec quatre informations relatives à l'al-hadith.

Donc le meilleur site est (إسلام ويب).

8. Conclusion :

Dans ce chapitre, nous avons conclu que la richesse des informations contenues dans l'exposé nous conduit à nous demander: comment extraire le maximum d'informations, la réponse peut être des moteurs de recherche suffisants et étendus. Donc nous avons traité des sites de recherche et les comparer pour extraire le meilleur en termes d'informations extraites.

Dans le chapitre suivant, nous présentons Les moteurs de recherche dédies.

Chapitre 02 :

Les moteurs de recherche dédient

1. Introduction :

Notre travail s'inscrit dans le domaine de la recherche d'information, pour concevoir un moteur de recherche dans un al-Hadith. Donc dans ce chapitre, nous expliquerons le fonctionnement de moteur de recherche en expliquant ses principaux composants.

À mesure que le volume d'informations augmente, il est nécessaire de développer des méthodes de recherche, seule l'indexation pouvant accélérer la recherche dans de très grands systèmes tels que le Web, car il anticipe la recherche en les extrayant et en les organisant par mots-clés.

Pour que les résultats de recherche soient satisfaisants, il faut bien calculer la pertinence des résultats par rapport à la requête, ceci est fait lors de l'interrogation. La question doit également pouvoir exprimer des questions simples ainsi que des questions complexes.

La qualité de la recherche est directement liée à la qualité de l'indexation et de la requête. Ces deux opérations peuvent être considérées comme le cœur du moteur de recherche. L'objectif de ce chapitre est de définir les principaux concepts de ce domaine, par étudier l'indexation, ses méthodes et ses étapes, puis nous expliquerons le processus de recherche et la notion de pertinence.

2. Définitions :

2.1. Mot clé : Mot ou groupe de mots choisi en vu de représenter le contenu d'un document, et de le retrouver lors d'une recherche documentaire. Il peut être issu du document (titre, texte, résumé,...). [Hanka1998]

2.2. Descripteur : Mot clé choisi parmi un ensemble de termes équivalents pour représenter sans ambiguïté un concept. Il fait en général partie d'un vocabulaire organisé ethiérarchisé de type 'thésaurus'. [Hanka1998]

2.3. 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. [Hanka1998]

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 type de langage de requette sont proposés dans la littérature.une requête est un ensemble de mots clés mais elle peut être exprimée en langage naturel, booléen ou graphique. [Hanka1998]

2.5. Pertinence : La pertinence est un mot qui signifie simplement renvoyé les informations considérées comme les plus utiles en haut de la liste des résultats. Bien que la définition soit

simple, faire en sorte qu'un programme calcule la pertinence ne soit pas une tâche triviale, principalement parce que la notion d'utilité est difficile à comprendre pour une machine. [Hanka1998]

3. Moteurs de recherche :

Un moteur de recherche est un logiciel qui permet de récupérer des ressources (pages Web, images, vidéos, fichiers, informations, etc.) liées à n'importe quel mot. Certains sites Web proposent principalement un moteur de recherche, appelé ensuite moteur de recherche, le site Web lui-même (Google, Yahoo, Bing ... sont des moteurs de recherche). [Nejjari2007]

Les moteurs de recherche ne s'appliquent pas uniquement au Web Certains lecteurs sont des logiciels installés sur des ordinateurs personnels. Il s'agit de moteurs de recherche de bureau, permettant de rechercher des fichiers stockés sur votre ordinateur tels que Exalead Desktop, Google Desktop, Copernic Desktop Search, etc. [Abulhajjaj2009]

Rien ne distingue un moteur de recherche « sémantique », d'un moteur de recherche « classique » comme Google. Même interface sobre, avec au centre de la page une fenêtre dans laquelle l'utilisateur entre sa requête.

4. Recherche en texte intégral :

La recherche en texte intégral est une technologie axée sur la recherche de documents correspondant à un ensemble de mots. La plupart des moteurs de recherche tels que Google et Yahoo! utiliser des moteurs de recherche en texte intégral au cœur de leur service. Les différences entre chacun d'eux sont des secrets de recettes (et parfois pas si secrets), tels que l'algorithme Google Page Rank.

Compte tenu de l'ensemble des mots (requête), la recherche en texte intégral a pour objectif principal de fournir un accès à tous les documents correspondant à ces mots. La numérisation séquentielle de tous les documents pour trouver les mots correspondants n'ayant pas une grande efficacité, le moteur de recherche de texte intégral est divisée en deux processus: indexer les informations dans un format efficace et rechercher les informations pertinentes à partir de cet index précompilé. [Bernard2009]

5. Processus d'indexation :

Pour que le coût de la recherche soit acceptable, il convient d'effectuer une étape primordiale sur la base documentaire. Cette étape consiste à analyser chaque document de la collection afin de créer un ensemble de mots-clés : on parle de l'′etape d'indexation. Ces mots-clés seront plus facilement exploitables par le système lors du processus ultérieur de recherche. L'indexation permet ainsi de

créer une représentation des documents dans le système. Son objectif est de trouver les concepts les plus importants du document (ou de la requête), qui formeront le descripteur du document. [Mechach2016]

5.1. Définition et rôle de l'indexation :

L'indexation est le processus consistant à en extrayant des mots directement à partir du document ou en assignant des mots à partir d'un vocabulaire contrôlé. Les termes de l'index sont ensuite présentés dans un ordre systématique. Les indexeurs doivent décider du nombre de termes à inclure et de leur précision. Ensemble, cela donne une profondeur d'indexation. [Lancaster2003]

L'indexation est souvent utilisée pour récupérer des informations. Mais peut également être utilisé dans d'autres domaines tels que la classification automatique de documents, la suggestion de mots clés, le calcul de termes simultanés, le résumé automatique, etc. [Abar2009]

5.2. Modèles d'indexation :

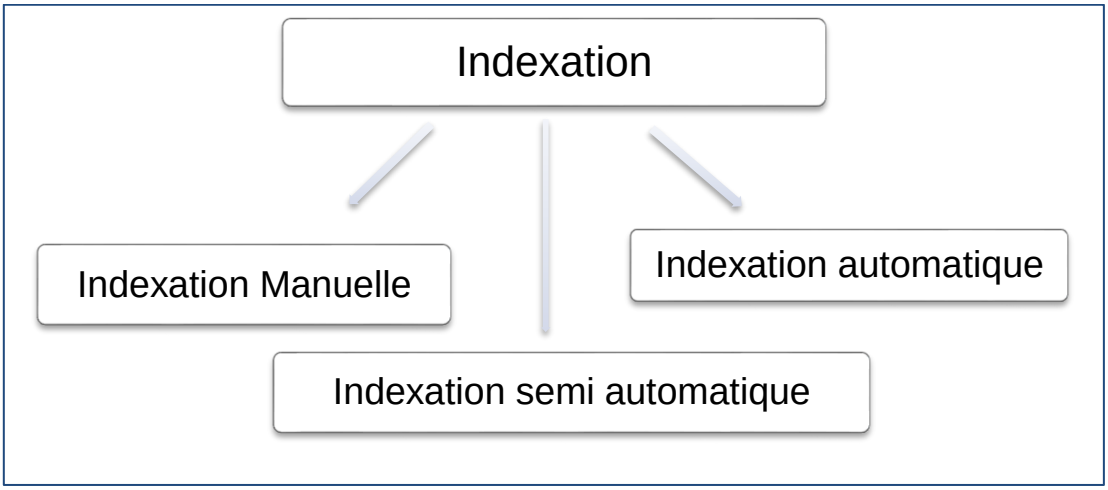


Figure 2.1 : Les modèles d'indexation

5.2.1. Indexation Manuelle : (indexation humaine)

Dans le cas d'une indexation humaine, c'est le documentaliste qui effectue l'analyse du document, pour identifier son contenu et construire une représentation de ce contenu. L'indexation manuelle est plus importante dans les réponses, car elle identifie des mots-clés plus spécifiques décrivant le document. [Sauvagnat2005]

Cependant, il à plusieurs inconvénients, il y a le problème du vocabulaire utilisé et de la dépendance à la connaissance de l'indexeur sur le sujet, c'est-à-dire qu'un même document peut être indexé de plusieurs manières (selon la vision de celui qui effectuel'indexation), et un indexeur à

deux moments différents peuvent avoir deux termes distincts pour représenter le même concept**.** [Amrouche2008]

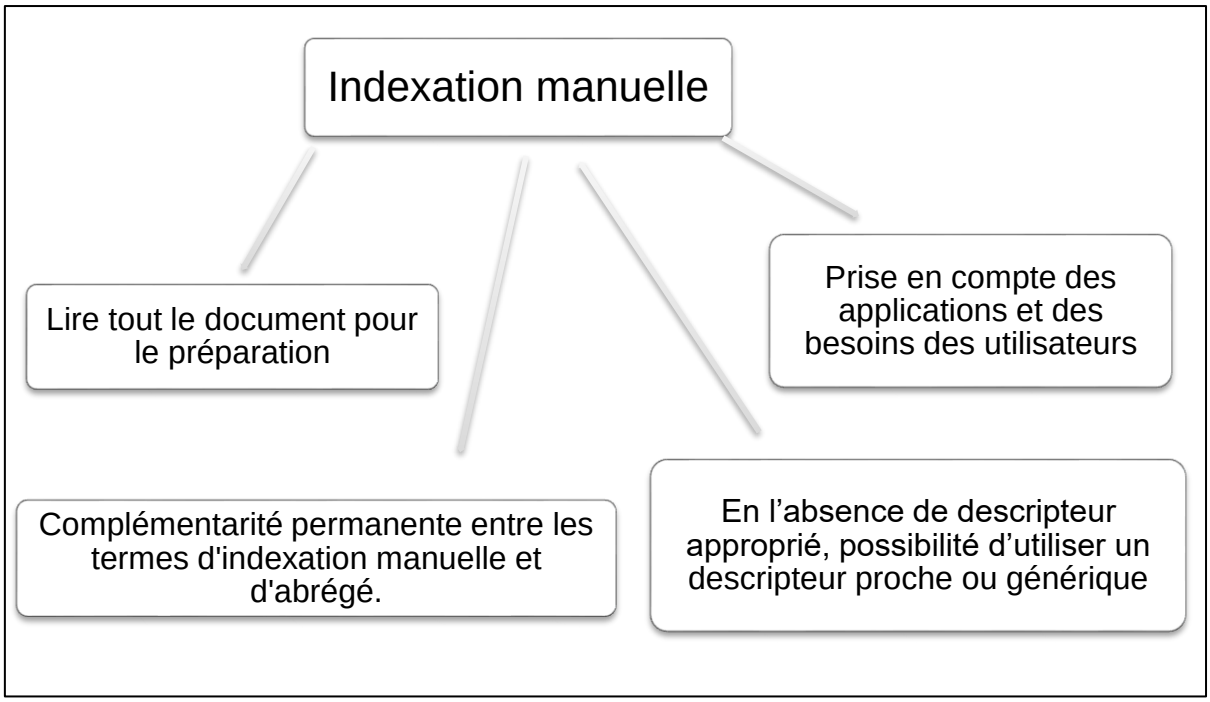


Figure 2.2 : Les points clés d'indexation manuelle.

5.2.2. Indexation automatique :

L'indexation automatique est l'opération qui consiste à faire reconnaître par l'ordinateur des termes figurant dans le titre, le résumé, le texte complet (s'il est enregistré avec la notice documentaire) et parfois même l'indexation humaine, et à employer ces termes, soit tels quels soit après conversion en d'autres termes équivalents ou conceptuellement voisins, pour en faire des critères incorporés dans le fichier de recherche et utilisables pour retrouver le document. [Chartron1989]

L'indexation automatique présente l'avantage d'une régularité du processus, car l'indexation automatique fournit toujours le même index pour le même document. Ce qui constitue une qualité du système, mais qui est différente de la justesse de l'indexation. En effet, l'indexation automatique pêche par son incapacité à interpréter un texte et son manque d'adaptation à de nouveaux vocabulaires. Il est impossible de trouver dans le document autre chose que ce que le système peut détecter. Par exemple, si le système n'a aucune connaissance lui permettant de lever les ambiguïtés des termes, il génèrera des erreurs d'interprétation du sens ce qui entraînera des incohérences dans la base des index. [Catherine2001]

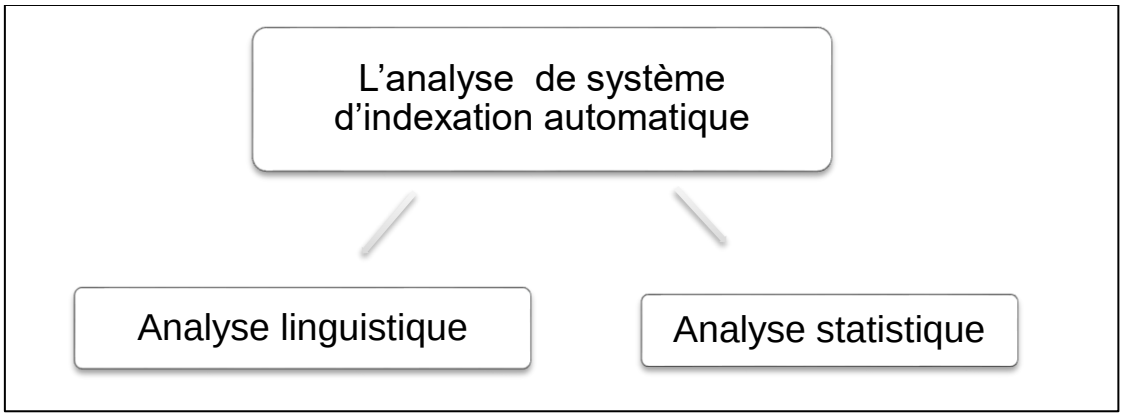


Figure 2.3 : Les méthodes de L'analyse de système d'indexation automatique

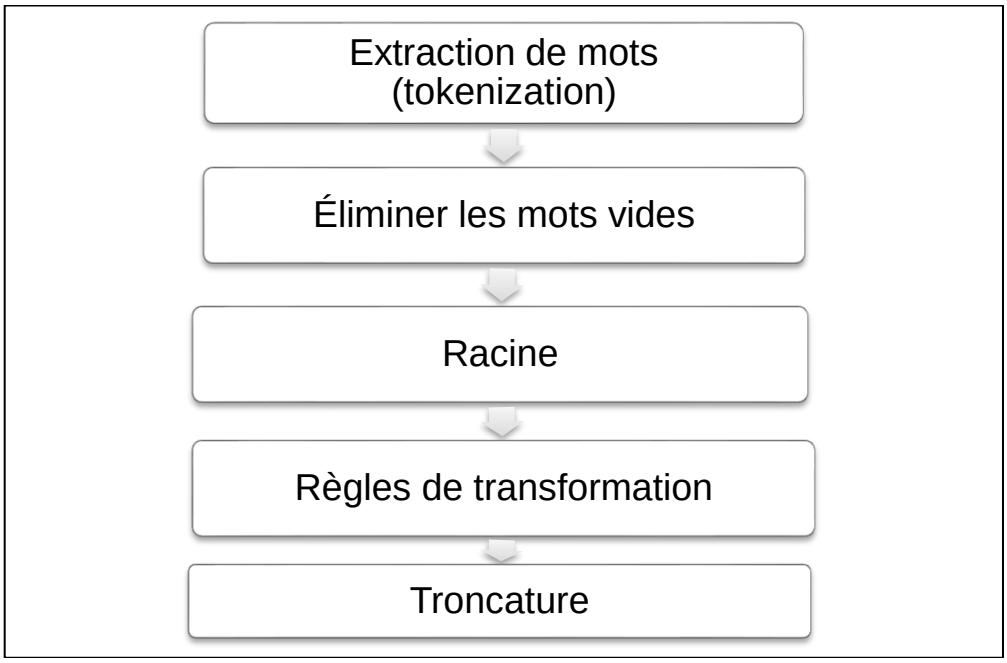


Figure 2.4 : Les étapes de L'indexation automatique

5.2.3. Indexation semi-automatique :

Les deux techniques précédentes peuvent être combinées, un premier processus automatique pour extraire les termes du document.

Toutefois, le choix final reste celui du spécialiste ou du bibliothécaire pour établir des relations sémantiques entre mots-clés et choisir les termes significatifs à l'aide d'un thésaurus ou d'une base de données terminologique, qui est une liste organisée de descripteurs (mots-clés) obéissant à des règles de terminologie spécifiques**.** [Abar2009, Sauvagnat2005]

5.3. Types d'index :

L'index est la sortie du processus d'indexation, il existe plusieurs types d'index en fonction de la technique utilisée et de la fonction souhaitée:

5.3.1. Index du document :

L'index de document conserve des informations sur chaque document, classées par ID du document.

Les informations stockées dans chaque entrée comprennent des données. Si le document a été analysé, il contient également un pointeur sur un fichier appelé informations du document, contenant l'URL et le titre. [Brin1998]

Le tableau suivant est une illustration simplifiée d'un index de document:Le tableau suivant est une illustration simplifiée d'un index de document:

Document ID	**Texte**	**Lien**
Document 1	محمد هو النبي صلى الله عليه وسلم	**/ex/doc1.txt**
Document 2	فاطمة هي بنت النبي محمد صلى الله عليه وسلم	**/ex/doc2.txt**

5.3.2. Index à terme :

L'index de transfert stocker une liste de mots pour chaque document. Ce qui suit est une forme simplifiée d'indexation:

Document ID	**Terme**
Document1	،محمد، هو، النبي، صلى، الله، عليه، و، سلم

Le principal objectif du développement d'un index de transfert est que lors de l'analyse de documents, il est préférable de stocker immédiatement les mots de chaque document.

L'index avant est trié pour le convertir en un index inversé. L'index avant est une liste de paires composée d'un document et d'un mot, regroupés par document. Convertir l'index en avant en un index inversé consiste à trier les paires par mot. Alors, donc l'index inversé est un index avant séparé par des mots. [Brin1998]

5.3.3. Index inverse:

De nombreux moteurs de recherche incluent un index inversé lors de l'évaluation d'une requête de recherche pour récupérer rapidement les documents contenant des mots dans la requête, puis les trier par pertinence. Comme l'index inversé stocke la liste des documents contenant chaque mot, le moteur de recherche peut utiliser l'accès direct pour rechercher les documents associés à chaque mot d'une requête afin de récupérer les documents qui répondent rapidement.

Le tableau suivant est une illustration simplifiée d'un index inversé :

Mot	Documents
محمد	Document 1, Document 3, Document 4
هو	Document 1, Document 3, Document
النبي	Document 2
صلى	Document 1, Document 5, Document 4
الله	Document 7, Document 3
عليه	Document 10
وسلم	Document 17

Cet index ne peut identifier que si un mot existe dans un document particulier car il ne stocke aucune information concernant la fréquence ou la position du mot. Cet index détermine les documents qui correspondent à une requête, mais ne les classe pas. Dans certains modèles, l'index inclut des informations supplémentaires telles que la fréquence ou les positions de chaque mot dans chaque document. Les informations de position permettent à l'algorithme de recherche d'identifier les mots adjacents pour appuyer la recherche par phrases. La fréquence peut être utilisée pour faciliter le calcul de la pertinence des documents par rapport à la requête. [Grossman2002]

5.4. Stockage d'index :

Le stockage des structures d'index est principalement caractérisé par la taille de l'index et l'organisation de ses éléments. Les structures d'index varient considérablement dans leur utilisation de taille étroitement liée à l'organisation des données dans l'index.

Cette organisation a un impact significatif sur le temps de recherche. De plus en plus d'éléments sont étroitement liés dans l'espace de stockage et sont moins recherchés en termes de temps de latence, c'est ce que l'on appelle le concept de localisation. Il est également important que l'index conserve la mémoire principale, évite les accès disque au système et limite l'accès à la recherche.

L'index idéal est celui qui occupe moins d'espace et réduit le temps de latence des recherches. [Dahak2006]

5.5. Phases d'indexation :

Le processus d'indexation comprend les phases suivantes:

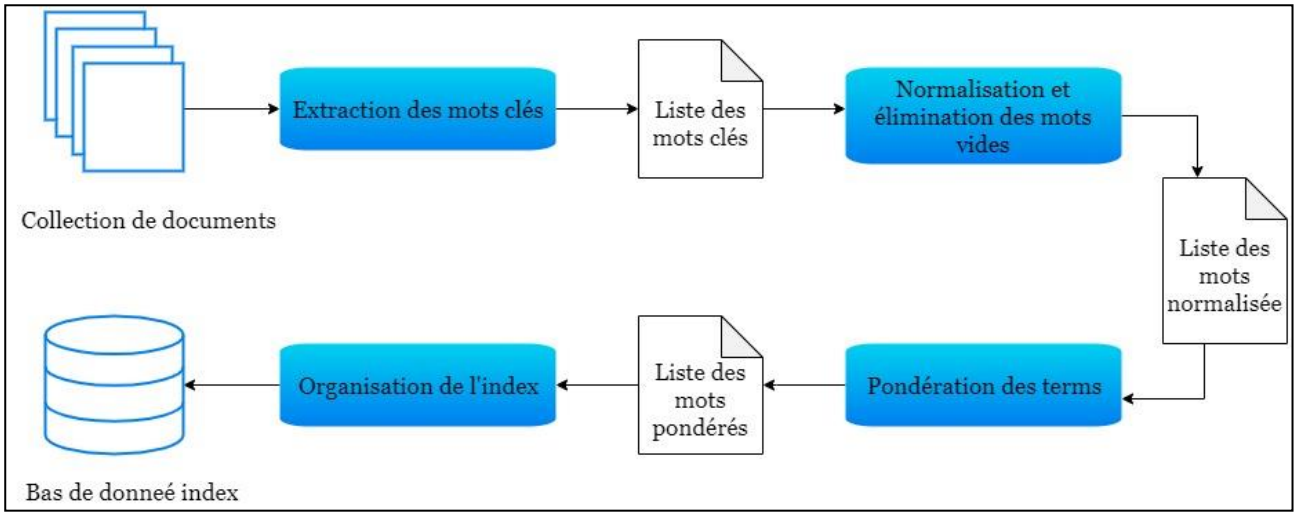


Figure 2.5 : Les phases de processus d'indexation

5.5.1. Tokénisation :

Est une phase qui peut sembler triviale au début, mais qui constitue néanmoins la base du reste des phases d'indexation. Par conséquent, cette phase doit être réalisée avec la plus haute qualité.

Certains systèmes de récupération utilisent une liste de mots-clés prédéfinis. Cette liste est conçue manuellement et, dans la plupart des cas, construite pour un sujet spécifique. Cette méthode permet de contrôler la taille de l'index. L'utilisation de l'extraction automatique de mots-clés ou l'utilisation d'une liste de mots-clés prédéfinis détermine le type d'indexation. [Dahak2006]

5.5.2. Normalisation :

Les index sont stockés que sous leurs formes normalisées, ce qui offre un gain de taille significatif, mais surtout, même si le traitement est effectué à la demande, il peut être beaucoup plus rapide et plus flexible dans la recherche, cette étape est aussi appelée "traitement morphologique de mots-clés". [Denoyer2004].

Cette phase peut également être enrichie du traitement syntaxique et sémantique des mots-clés. La première consiste à identifier et à grouper un ensemble de mots dont la signification dépend de leur union. [Dahak2006]

5.5.3. Elimination des mots vides :

Cette phase est d'une certaine importance car elle constitue un facteur de grande influence sur la précision de la recherche. Le fait de ne pas supprimer les mots vides provoque inévitablement du bruit. L'élimination des mots vides qui sont des mots du langage courant et qui contiennent peu d'informations sémantiques doit être à la fois dans l'indexation et dans la requête (suppression des mots vides de la requête). [Dahak2006]

5.5.4. Pondération :

Cette étape dépend entièrement du modèle de récupération d'informations utilisé. Il définit l'importance d'un terme dans un document donné. [Dahak2006]

En général, la plupart des formules de pondération sont construites en combinant deux facteurs. Un facteur de pondération local mesurant la représentativité locale d'un terme dans le document et un facteur de pondération global mesurant la représentativité globale d'un terme par rapport à la collection de documents [Amrouche2008].

5.5.4.1. Pondération locale :

La pondération locale prend en compte les informations locales du terme qui ne dépendent que du document. C'est typiquement une fonction de la fréquence d'occurrence du mot dans le document, notée **TF**(TermeFréquence).

Un terme qui apparaît fréquemment dans un document est considéré comme pertinent pour décrire son contenu**.** [Dahak2006]

5.5.4.2. Pondération globale :

Le poids total mesure l'importance d'un terme dans tous les documents. Il vise à représenter son caractère discriminatoire ou, en d'autres termes, sa capacité à distinguer un document. En fait, un terme figurant dans peu de documents est considéré comme plus discriminatoire et devrait être

privilégié par rapport à un terme trouvé dans de nombreux documents. Le calcul de la pondération globale est basé sur le nombre de documents dans lesquels un terme apparaît. L'un des plus utilisés est **idf** (Inverse Document Fréquency), représenté par la formule suivante:

$$\mathbf{Idf=log_2(\frac{N}{N_i})}$$

Tels que est le nombre de documents contenant le mot (i) et (N) le nombre total de documents.

La valeur **(tf * idf)** donne une bonne approximation de l'importance d'un terme dans le document, en particulier dans le corpus de documents de taille similaire. [Dahak2006]

6. 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**.** [Mechach2016]

Ce processus fournit une requête interne. Après cette étape de En interprétant la requête, le modèle de correspondance calcule la correspondance entre la requête interne et chaque document de l'index. Cecompte créé par la fonction de nomination conduisait traditionnellement à une liste ordonnée de documents. À ce niveau, il convient de comparer les connotations (non égales) entre les concepts du document et ceux du document.

La comparaison de la requête et du document conduit rarement à des équations strictes, Mais pour les équations partielles: Le document n'est qu'une partie de la requête. Le premier document de la liste renvoyée par le système est celui que le système considère comme le plus pertinent, le document le plus approprié pour la requête, toujours selon au système. Le document final est le document considéré comme le moins important. Cette notion de commodité repose sur la proximité entre les besoins exprimés par l'utilisateur et les résultats fournis par le système. [Abbes2008]

6.1. Notion de pertinence :

Voyons quelques définitions pertinentes. La pertinence est:

- la correspondance entre un document et une requête, une mesure d'informativité du document à la requête.
- un degré de relation (chevauchement, relativité, …) entre le document et la requête.
- un degré de la surprise qu'apporte un document, qui a un rapport avec le besoin de l'utilisateur.
- une mesure d'utilité du document pour l'utilisateur.

Même dans ces définitions, les concepts utilisés (informatique, relativité, surprise ...) restent très vagues car les utilisateurs ont des besoins très différents. Ils ont des critères très différents pour

juger de la pertinence du document. La notion d'importance couvre donc un très large éventail de normes et de relations**.** [Mechach2016]

6.2. Les modèles de recherche d'information :

Un modèle de RI a pour rôle de fournir une formalisation du processus de RI et un cadre théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuelle développé dans la littérature.

Ces modèles ont en commun le vocabulaire d'indexation basé sur le formalisme mots clés et diffèrent principalement par le modèle d'appariement requête-document.

Le vocabulaire d'indexation $V = \{t_i\}$, $i \in \{1, ..., n\}$ Est constitué de (n) mots ou racines de mots qui apparaissent dans les documents.

Un modèle de RI est défini par un quadruplet (D, Q, F, R (q, d)) **: où**

- (D) est l'ensemble de documents
- (Q) est l'ensemble de requêtes
- (F) est le schéma du modèle théorique de représentation des documents et des requêtes
- R (q, d) est la fonction de pertinence du document (d) à la requête (q) Nous présentons dans la suite les principaux modèles de RI : le modèle booléen, le modèle vectoriel et le modèle probabiliste. [Mechach2016]

6.2.1. Modèle booléen :

Est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. Un exemple de représentation d'un document est comme suit : d = $t_1 \wedge t_2 \wedge t_3 ... \wedge t_n$.

Une requête est une expression booléenne dont les termes sont reliés par des opérateurs logiques (OR, AND, NOT) permettant d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. Un exemple de représentation d'une requête est comme suit : " q = $(t_1 \wedge t_2) \vee (t_3 \wedge t_4)$ ".

La fonction de correspondance est basée sur l'hypothèse de présence / absence des termes de la requête dans le document et vérifie si l'index de chaque document dj implique l'expression logique de la requête (q). Le résultat de cette fonction est donc binaire est décrit comme suit : RSV (q, d) = {1,0}. [Mechach2016]

6.2.2. Modèle vectoriel :

Dans ces modèles la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel. Le modèle vectoriel représente les documents et les requêtes par des vecteurs d'un espace à (n) dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation.

L'index d'un document$d_j$djest le vecteur =(W_{1j}, W_{2j}, W_{3j}…..W_{kj}), oùW_{kj}∈ [0 1] dénote le poids du terme (t_k) dans le document (d_j).

Une requête est également représentée par un vecteur = (W_{1q}, W_{2q} , W_{3q} ,.., W_{nq}), où W_{kq}est le poids du terme(t_k)dans la requête (q).

La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs : RSV (q, d) =cos (q, d**).** [Mechach2016]

6.3. Processus de recherche :

Les différentes étapes du processus de RI, sont représentées schématiquement par le processus en U (Figure07).

La figure illustre particulièrement :

- Les notions de documents et de requêtes qui sont des conteneurs d'informations.
- Les opérations d'analyse, d'indexation et d'appariement qui permettent globalement de traiter la requête dans le but de sélectionner des documents à présenter à l'utilisateur**.** [Bernard2009]

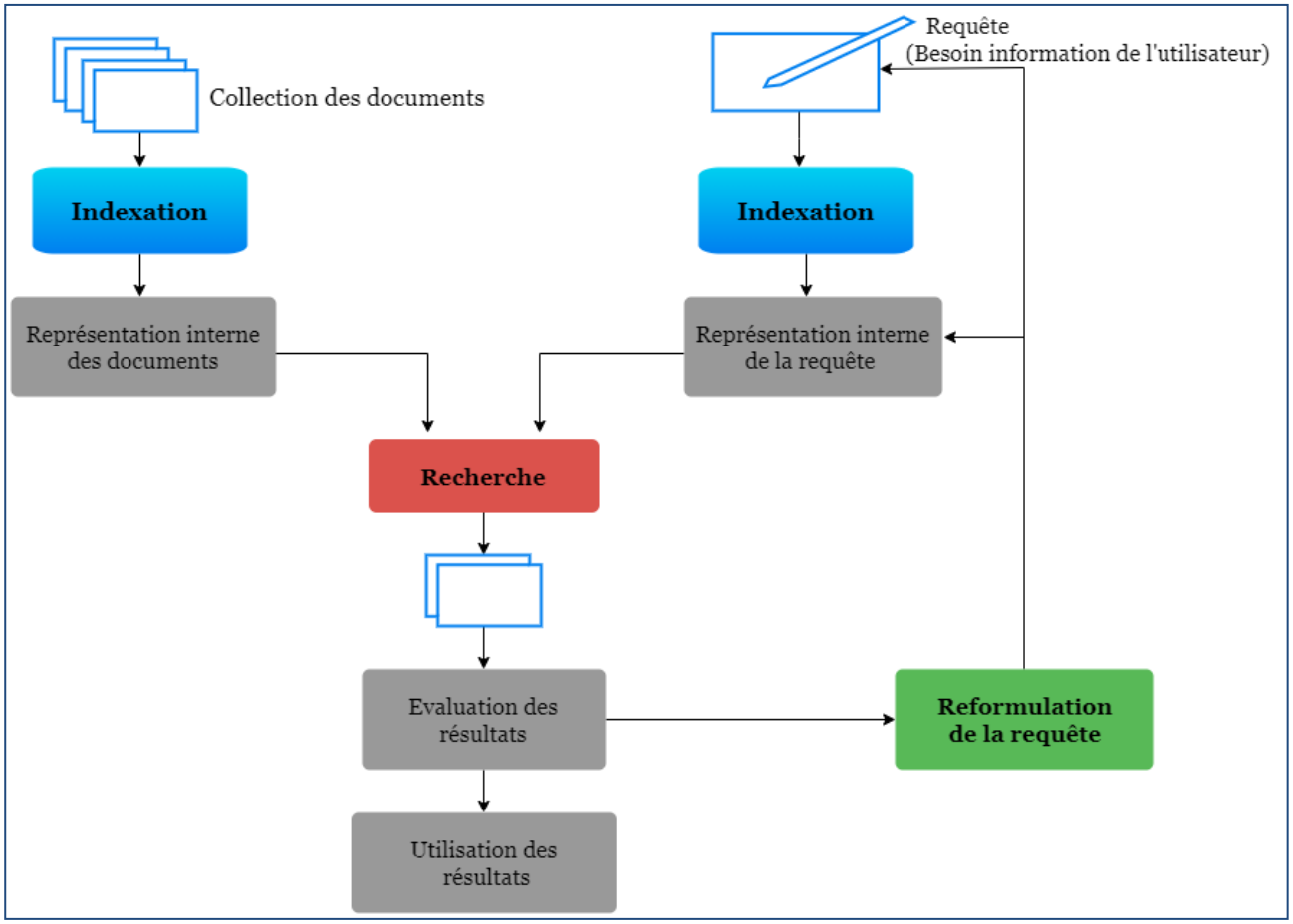


Figure 2.6 : Processus de recherche

7. Recherche sémantique :

La recherche sémantique est une méthode de recherche dans laquelle une requête de recherche vise non seulement à rechercher des mots-clés, mais également à déterminer le sens et le contexte prévus des mots utilisés par la personne à rechercher.

La recherche sémantique fournit des résultats de recherche plus significatifs en évaluant et en comprenant le terme de recherche et en recherchant les résultats les plus pertinents sur un site Web, une base de données ou un autre entrepôt de données.

Les principaux moteurs de recherche sur le Web, tels que Google et Bing, incorporent des éléments de recherche sémantique.

Un moteur de recherche intelligent prend en compte plusieurs facteurs pour fournir les requêtes de recherche les plus pertinentes et utiles, notamment :

– **Mode actuelle:** Si l'élection du président venait juste de s'achever dans le pays et que quelqu'un recherchait "Qui est le nouveau président", le système de recherche sémantique devrait être en mesure de comprendre la requête et de donner des résultats pertinents en fonction de la tendance et des actualités actuelles.

– **Lieu de la recherche:** si une personne recherche 'quelle est la température, le moteur de recherche sémantique doit être en mesure de fournir des résultats basés sur le emplacement de la recherche. Si la personne cherche en Californie, les résultats de la recherche doivent inclure la température actuelle en Californie.

– **Variations de mots dans la recherche sémantique:** La recherche sémantique doit prendre en compte les temps, le pluriel, le singulier, etc. et fournir des résultats de recherche pertinents pour toutes les variations sémantiques des mots. Par exemple, des mots comme chien, chien, chien, etc.

– **Recherche sémantique et synonymes:** un moteur de recherche sémantique devrait pouvoir comprendre les synonymes et donner plus ou moins les mêmes résultats de recherche synonymes du mot recherche par les utilisateurs. Par exemple, essayez de rechercher «plus grande montagne» et «plus haute montagne». Vous obtiendrez à peu près les mêmes résultats puisque les deux signifient la même chose dans cette requête, même si «plus gros» et «plus haut» peuvent signifier différentes choses dans différents cas. [Web-RS]

8. Conclusion :

Dans ce chapitre, l'étude portait sur le mécanisme de travail des moteurs de recherche et des systèmes de recherche d'informations, fondée sur l'indexation en raison de son importance. En effet, il s'agit de l'étape la plus importante du processus de recherche car elle permet l'extraction et le traitement de mots-clés.

La phase de recherche n'offre pas uniquement l'interaction entre les utilisateurs et le système, mais calcule également le pourcentage de correspondance entre la requête et les documents afin de fournir les résultats les plus pertinents.

Chapitre 03 :

Les fonctionnalités ElasticSearch

1. Introduction :

Nous avons a besoin de stocker de plus en plus d'informations, et ces rechercher de façon toujours plus rapide et efficace. Une recherche dans une base de données relationnelle peut paraître simple et rapide si la requête est bien construite avec les jointures nécessaires entre les tables. Mais malheureusement, plus cette base de données va grossir, plus les requêtes seront longues à exécuter, et il ne peut pas effectuer de recherche en texte intégral, gérer des synonymes et trier les documents par pertinence, ce à cette problématique qu'elasticSearch essaye de répondre.

Dans ce chapitre, nous expliquerons le fonctionnement d'elasticsearch en expliquant ses principaux composants.

2. Qu'est-ce qu'Elasticsearch ? :

Elasticsearch est un projet de serveur de recherche open source lancé par Shay Banon et publié en février 2010. Il est ensuite devenu un acteur majeur dans le domaine des solutions de recherche. De plus, en raison de sa nature distribuée et de ses capacités en temps réel, de nombreuses personnes l'utilisent comme base de données de documents. [Rafal2013]

3. Avantage :

– Elasticsearch est développé sur Java, ce qui le rend compatible sur presque toutes les plateformes.

– Elasticsearch est temps réel, en d'autres termes après un deuxième, le document ajouté est consultable dans ce moteur.

– Elasticsearch est distribuée, ce qui facilite l'échelle et l'intégration dans toute une grande organisation.

– Elasticsearch utilise des objets JSON comme réponses, ce qui permet d'invoquer le serveur elasticsearch avec un grand nombre de langues de programmation différentes.

– Elastisearch prend en charge presque tous les types de documents, sauf ceux qui ne prennent pas en charge le rendu de texte.

– La création de sauvegardes complètes est facile en utilisant le concept de passerelle, présent dans elasticsearch. [Robin2016]

4. Inconvénients :

Elasticsearch travaille uniquement avec JSON et n'est donc pas compatible directement avec les formats de données CSV et XML: il faut une pré-phase de transformation vers JSON. [Robin2016]

5. Fonnctionnements d'Elasticsearch:

5.1. Presentation d'Apache Lucene:

Lucene est une sous-structure open source de moteur de recherche qui a été utilisé développé par Apache. Il constitue la sous-structure d'elasticsearch parmi les moteurs de recherche qui conservent leur popularité dans la technologie actuelle. Lucene a été développé essentiellement avec Java. Apache Lucene est une bibliothèque qui a été développée pour remplir les processus de recherche de texte intégral. Les données à indexer sont envoyées et elles sont indexées par Lucene sur le système de fichiers. Il permet l'indexation sur la zone autant que demandé. Lorsque le processus de recherche est effectué, il effectue une recherche sur le système de fichiers indexé. Il présente des résultats plus performants que les processus de recherche de texte intégral dans le SGBDR. C'est une bibliothèque idéale pour indexer une quantité énorme de données et effectuer une recherche. [Kuc2013]

5.2. Architecture générale :

- **Index :** est le lieu logique où elasticsearch stocke les données logiques, de manière à ce qu'elles puissent être divisées en éléments plus petits, et peut être répliqué sur zéro ou plusieurs Secondary Shards. [Kuc2013]
- **Cluster et Noeud:** Elasticsearch peut fonctionner en tant que serveur autonome à recherche unique. Néanmoins, pour pouvoir traiter de grands ensembles de données et pour obtenir une tolérance aux pannes et une haute disponibilité, elasticsearch peut être exécuté sur de nombreux serveurs coopérants. Ensemble, ces serveurs sont appelés un cluster et chaque serveur qui le constitue est appelé un nœud. [Kuc2013]
- **Document :** c'est l'objet qui permet de représenter une donnée (au sens NoSQL du terme). Par contre, pour ce faire, il est nécessaire de penser recherche et de penser document en oubliant toute notion de SGBDR et donc de jointure. [Kuc2013]
- **Shard :** Lorsque nous avons un grand nombre de documents, nous pouvons arriver à un point où un seul nœud peut ne pas suffire. Dans un tel cas, les données peuvent être divisées en parties plus petites appelées fragments (où chaque fragment est un index Apache Lucene distinct). Chaque fragment peut être placé sur un serveur différent et ainsi vos données peuvent être réparties entre les nœuds du cluster. Lorsque vous interrogez un index créé à partir de plusieurs fragments, elasticsearch l'envoie à chaque fragment pertinent et fusionne le résultat de sorte que votre application ne connaisse pas les fragments. De plus, plusieurs fragments peuvent accélérer l'indexation. [Kuc2013]

– **Réplicas :** Pour augmenter le débit des requêtes ou atteindre une haute disponibilité, vous pouvez utiliser des répliques de fragments. Une réplique est juste une copie exacte de la partition, et chaque partition peut avoir zéro réplique ou plus. En d'autres termes, Elasticsearch peut avoir plusieurs fragments identiques et l'un d'eux est automatiquement choisi comme emplacement où les opérations qui modifient l'index sont dirigées.Ce fragment spécial s'appelle un fragment primaire et les autres sont appelés des fragments de réplique. Lorsque le fragment principal est perdu (par exemple, si un serveur contenant les données de fragment est indisponible), le cluster va promouvoir le réplica en tant que nouveau fragment principal. [Kuc2013]

5.3. Format fondamentaux:

5.3.1. JSON:

Est un format léger d'échange de donnée. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou générale par des machines.

JSON se base sur deux structures :

– Une collection de couples nom / valeur (aussi appelé objet). Formalisé en JSON par une {à gauche et une} à droite. Chaque clé est suivi de : et les couples clé/valeur sont séparés par (,).

– Une liste de valeurs ordonnées (aussi appelé tableau). Formalisé en JSON par une [à gauche et une] à droite. Les valeurs sont séparés par des (,).

Une valeur peut être soit une chaîne de caractères entre guillemets, soit un nombre, soit true ou false ou null, soit un objet soit un tableau. Ces structures peuvent être imbriquées. [Robin2016]

5.3.2. API Rest :

Elasticsearch s'utilise avec l'API RESTful permettant d'effectuer tous types d'opération. Il supporte les méthodes HTTP (GET, PUT, POST et DELETE) **:**

– **GET:** est équivalent à une requête SQL Select. Il permet d'interroger la base de données et de récupérer les données.

– **PUT:** est équivalent à une requête SQL Insert. Il permet d'ajouter des données à la base.

– **POST:** est équivalent à une requête SQL Update. Il permet de modifier des données déjà enregistrées dans la base.

– **DELETE**: il permet de supprimer des données. [Razvan2016]

5.4. Rôle de logstash et kibana :

5.4.1. Logstash :

Logstash est un outil très puissant et très polyvalent car il est capable de traiter tout ce qui ressemble de près ou de loin à du texte. [Antoine2015]

Logstash se compose de 3 zones :

– **Les entrées :** Logstash accepte pratiquement tous les fichiers possibles en entrée, par exemple **:** Sqlite, file, jdbc. [Web_AN]

– **Les filtres** : Le GROS point for de logstash, c'est sa capacité à pouvoir filtrer tous les types possibles de données. Il existe de nombreux plugins qui permettent d'extraire, d'analyser et de nettoyer les données, d'ajouter et de supprimer des champs, de détecter du texte en fonction d'un pattern et de le parseur**.** [Web_AN]

– **Les sorties :** L'output peut servir à passer d'un mode de stockage à un autre. Par exemple, il est possible de convertir un fichier **csv** vers une base de données elasticsearch. [Antoine2015]

5.4.2. Kibana :

Kibana est utilisé pour créer des visualisations et créer des tableaux de bord pour les index présentés dans elasticsearch. Fondamentalement, c'est un plugin open source pour elasticsearch. [Antoine2015]

Kibana se compose de plusieurs onglets :

– **Discover :** Cet onglet permet de faire des recherches via la barre dédiée et de voir les résultats bruts stockés dans la base elasticsearch. [Antoine2015]

– **Visualize** : Cet onglet permet de réaliser différents types de graphes à partir d'une requête enregistrée ou à partir d'une nouvelle requête. [Antoine2015]

– **Dashboard :** Cet onglet permet de regrouper plusieurs graphes réalisés au préalable dans l'onglet visualise sur une même page. [Antoine2015]

6. Indexation d'un document:

– Elasticsearch permet de créer des indexes et ajouter des documents, par l'utilisation de la méthode PUT, comme suivante : [Kuc2013]

Curl -XPUT 'http: // localhost: 9200 / [index] / [type] / [id] / [_action]'

– La mise à jour de documents dans l'index est une tâche plus compliquée, mais Elasticsearch permet à ce d'extraire le document, ses données du champ _source, supprimer l'ancien document, appliquer les modifications apportées au champ _source, puis l'indexé en tant que nouveau

document via un script donné en tant que paramètre de demande de mise à jour, comme suivante:[Kuc2013]

Curl -XPOST 'http: // localhost:9200 / [index] / [type] / [_action]'-d ' {}'

– Après avoir ajouté tous nos documents, elasticsearch permet d'obtenir un document en utilisant son identification, avec la commande suivante : [Kuc2013]

Curl -XGET 'http: // localhost: 9200 / [index] / [type] / [id] '

– Nous avons déjà vu comment elasticsearch permettre de créer et récupérer des documents et comment les mettre à jour. Il n'est pas difficile de deviner qu'il permet aussi de suppression ce document avec la requête DELETE, comme suivante: [Kuc2013]

Curl -XDELETE 'http: // localhost: 9200 / [index] / [type] / [id] '

6.1. Recherché un document :

6.1.1. Recherche simple:

La recherche la plus simple consiste à demander tous les n-uplets enregistrés, à faire un **GET** sur **_search** sans filtre. [Dixit2017]

6.1.2. Recherche avec une query-DSL:

Query-DSL est une interface JSON fournie par elasticsearch pour écrire des requêtes au format JSON. Il vous permet d'écrire toute requête que vous pourriez écrire dans Lucene. Les requêtes peuvent être aussi simples que faire simplement correspondre des termes simples, ou elles peuvent être très complexes. [Dixit2017]

6.1.2.1. Requête:

L'objet de requête contient toutes les requêtes devant être transmises à elasticsearch afin de rechercher tous les documents appartenant à un champ de recherche. Il existe deux grandes catégories de requêtes :

– **Requêtes de recherche en texte intégral**: il s'agit des requêtes qui s'exécutent généralement sur des champs de texte. Ces requêtes comprennent le mappage de champ et, en fonction du type de champ et de l'analyseur utilisés pour ce champ et cette requête, le texte passe par une phase d'analyse (similaire à l'indexation) pour rechercher les documents pertinents. [Paro2015]

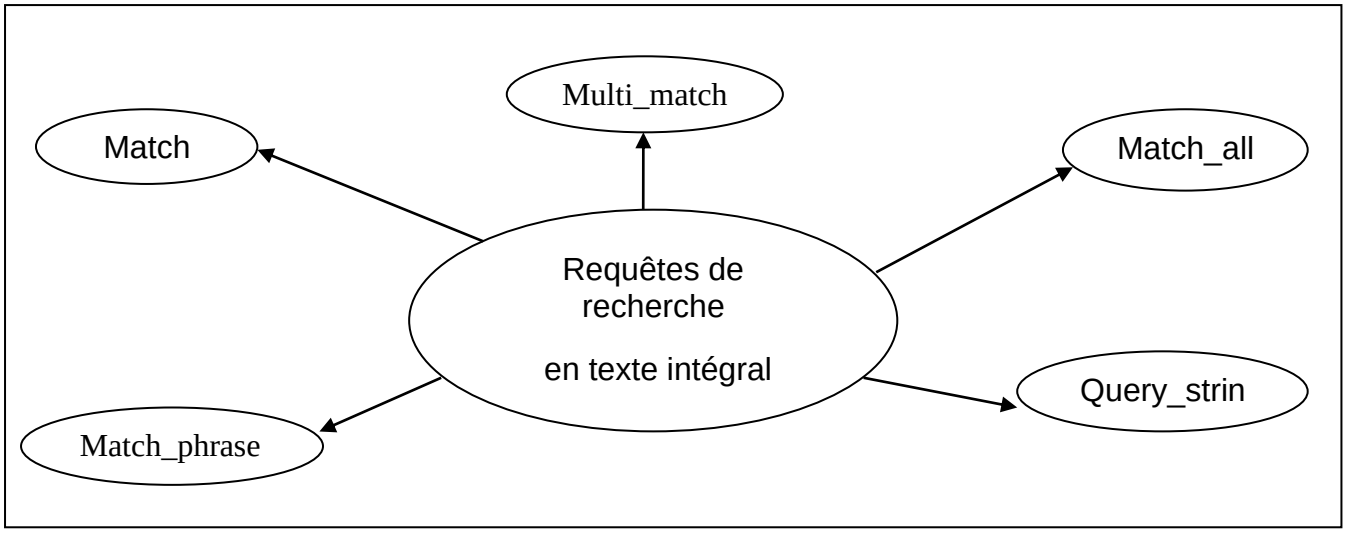


Figure 3.1 : Les requêtes les plus importantes dans la recherche en texte intégral

– **Requêtes de recherche basées sur des termes:** contrairement aux requêtes en texte intégral, les requêtes basées sur des termes ne passent pas par un processus d'analyse. Ces requêtes sont utilisées pour faire correspondre les termes exacts stockés dans un index inversé. [Paro2015]

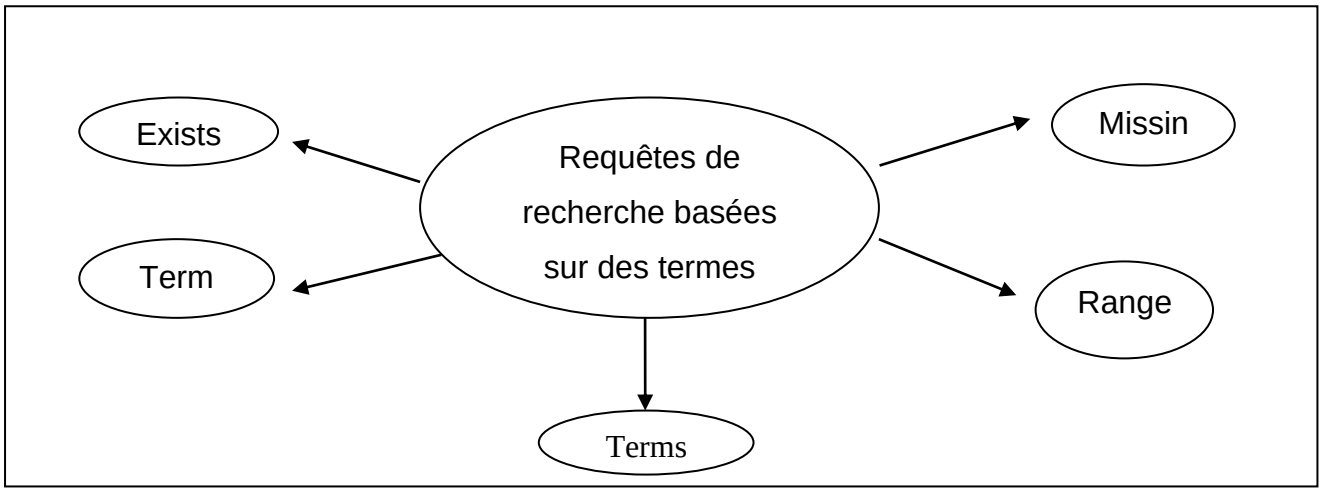


Figure 3.2 : Les requêtes les plus importantes dans la recherche basée sur des termes

6.1.2.2. Filters:

Le terme filtre est très puissant car il vous permet de définir la stratégie à utiliser pour filtrer les termes. Les stratégies sont passées dans le paramètre d'exécution:

– **Bool :** Ce paramètre génère une requête de terme pour chaque terme, puis crée un filtre booléen à utiliser pour filtrer les termes. Cette approche vous permet de réutiliser le terme filtres requis pour le filtrage booléen, ce qui augmente les performances si les filtres de sous-terme sont réutilisés.

– **And :** Ce paramètre est similaire au paramètre bool, mais les sous-requêtes du filtre de filtrage sont encapsulées dans un filtre AND.

– **OR :** ce paramètre est également similaire au paramètre bool, mais les sous-requêtes du filtre de filtrage sont encapsulées dans un filtre OR. [Paro2015]

6.1.3. Aggregation:

Dans le développement de solutions de recherche, non seulement les résultats sont importants, mais ils nous aident également à améliorer la qualité et la recherche. ElasticSearch fournit un outil puissant pour atteindre ces objectifs: les agrégations. Les agrégations sont principalement utilisées pour fournir des données supplémentaires aux résultats de la recherche afin d'améliorer leur qualité ou de les compléter avec des informations supplémentaires. [Paro2015]

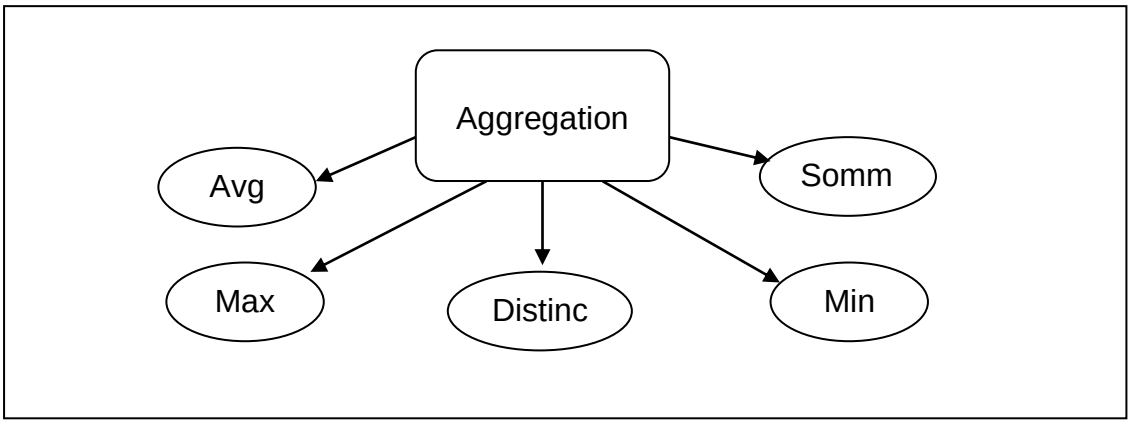


Figure 3.3 : Les types d'agrégation

7. Conclusion :

Dans ce chapitre, nous avons vu les fonctionnements de bases relatives à elasticSearch, c'est à dire l'indexation, la mise à jour, la suppression et la recherche de documents. Elasticsearch capable de gérer de grandes quantités de données et rechercher-les dans son environnement clustérisé.

Conception et réalisation

Chapitre 04 : Conception

1. Introduction :

Le hadith est la deuxième source des règles de l'islam, et il est extrêmement important de considérer al-hadith et d'étudier toutes les informations qui s'y rapportent pour savoir si al-hadith est vrai ou non.

En raison de l'importance de la recherche dans al-hadith qui se reflète dans la mémorisation de la Sunna et a hérité de la prophétie, nous avons suggéré de concevoir un système pour récupérer des informations qui répondent aux besoins spéciaux al-hadith, mais pour atteindre cet objectif, nous devons d'abord classer les fonctionnalités de recherche possibles et utiles.

Ce chapitre présente également l'architecture et divers schémas liés à la conception de notre système.

2. Difficultés de recherche dans l'al-hadith :

Pour explique la vision de la problématique, nous décrivons les difficultés de la recherche dans Al-hadith comme suivant:

- Premièrement, en tant que besoin général de recherche.
- Deuxièmement, comme un défi pour la recherche en arabe.

Commencez par expliquer le premier point, la recherche dans la théorie d'aL-hadith présente les mêmes difficultés que la recherche dans tout autre document. Où la recherche dans les documents a traversé par étapes de développement. En début, la recherche était séquentielle en fonction d'un mot clé spécifique avant que des expressions régulières ne soient fournies. Il à mettre la recherche en texte intégral pour éviter les restrictions de recherche séquentielle sur les documents volumineux. La recherche en texte intégral introduit de nouveaux mécanismes pour l'analyse de texte. L'agrégation de statistiques fait désormais partie du processus de recherche, ce qui permettre à améliorer le classement des résultats et des suggestions. La recherche tend à adopter une approche dans laquelle la précision de la recherche est améliorée grâce à la compréhension de l'intention du chercheur et de la signification contextuelle des termes tels qu'ils apparaissent dans l'espace de données de recherche pour générer des résultats plus pertinents. Pour plus d'expérience utilisateur, recherchez les moteurs tentent d'améliorer le comportement d'affichage des résultats en les catégorisant en fonction de leur pertinence dans les documents, les critères de tri, la mise en évidence des mots clés et des pages, le filtrage et le développement.

Deuxièmement, la langue d'al-hadiths est l'arabe classique. L'arabe est une langue spécifique en raison de la morphologie et de la dictée, et cela doit être pris en compte dans les étapes de l'analyse de texte. Par exemple, la formation de lettres (en particulier de Hamza -ء-), de prononciation, et de

différents niveaux de sortie, de types de dérivation, etc. Ceci doit être pris en compte dans les fonctionnalités de recherche, par exemple: Les expressions régulières représentent mal les lettres arabes. Le manque de prononciation crée une certaine ambiguïté dans la compréhension des mots:

Le mot « الملك » pourrait être compris comme « المَلَك » (l'ange), « المَلِك » (le roi), « المُلْك » (le royaume). [Albawwab2009]

Expliquez à cette étape les défis spécifiques que nous rencontrons dans la recherche en fonction des caractéristiques d'AL-Hadith :

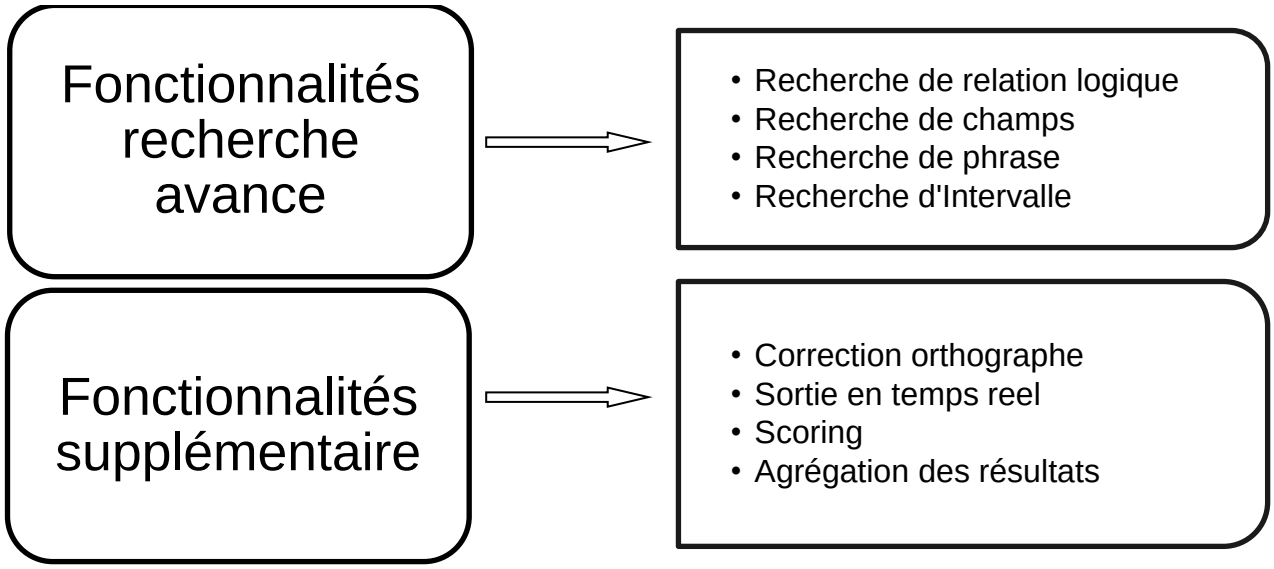
- **structure principale**: ChainNarrator (sanad السند), Matn (المتن).
- **Structure informelle** : Fiabilité (hokm حديث صحيح, حديث حسن), author (مخرج الحديث), Narrator (راوي الحديث), collection (source مصدر الحديث), Book (كتاب الحديث), Chapter (باب الحديث).
- **Révélation**: lieu, calendrier, raison, contexte, etc.

Les utilisateurs peuvent avoir besoin de rechercher, filtrer ou agréger les résultats en fonction de l'une de ces structures. Il existe de nombreuses sciences liées aux hadiths, appelées sciences de hadiths: interprétation, traduction, etc.

Ensuite, nous suggérerons une ventilation préliminaire des fonctionnalités de recherche que nous avons inspirées de manière problématique.

3. Classification des fonctionnalités de recherche d'al-hadith:

Pour faciliter l'inclusion des fonctionnalités de recherche, nous les avons regroupées dans plusieurs catégories en fonction de leurs objectifs :



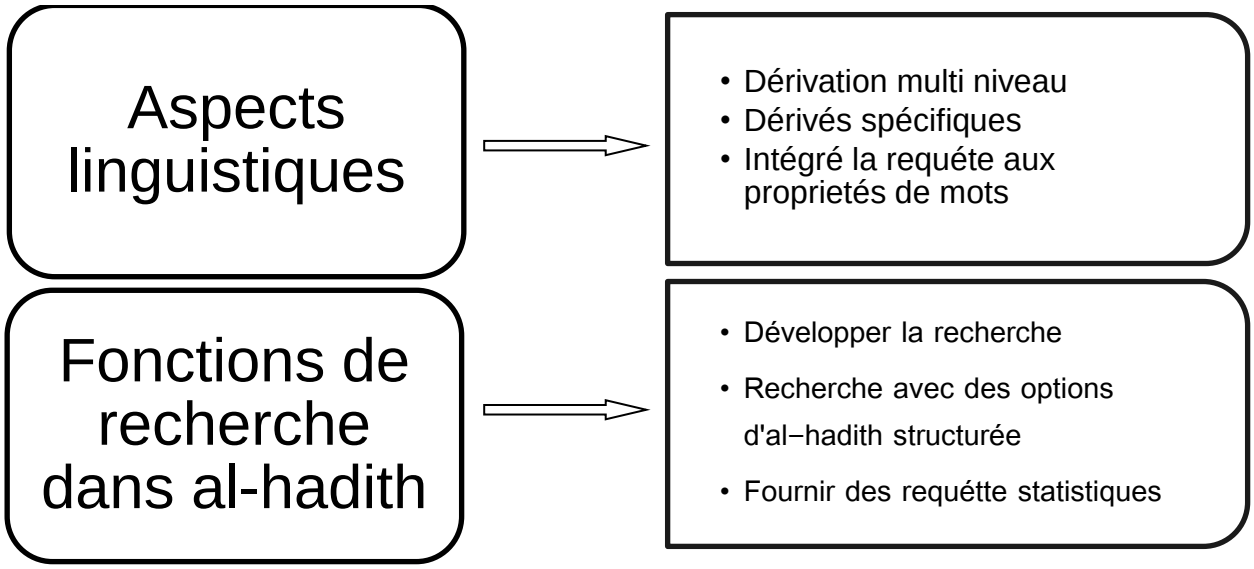


Figure 4.1 : Classification des fonctionnalités de recherche d'al-hadith

3.1. Fonctionnalités de recherche avancées :

– **Recherche de relations logiques :** implique la présence ou l'absence d'un mot clé. Les relations les plus connues sont: (AND) pour l'appariement, (OR) pour la séparation et (NOT) pour l'exception. Les relations peuvent être regroupées à l'aide de parenthèses. Par exemple : (الصلاة AND_NOT الزكاة) AND fiabilite = حديث صحيح, pour trouver toute des al-hadiths al-sahiha contenant le mot (الصلاة), à l'exception de celui contenant le mot (الزكاة).

– **Recherche de champ:** utiliser le nom de champ dans la requête pour rechercher un champ spécifique, est utile pour rechercher des informations plus diverses telles que chapter, book, collection, etc.

– **Recherche de phrase :** type de recherche permettant aux utilisateurs de rechercher des documents contenant une phrase ou une phrase spécifique. Par exemple : ("الحمد لله "), pour faire correspondre exactement les mots si et seulement s'ils sont adjacents et sous la même forme.

– **Recherche d'intervalle (par range):** utilisé pour rechercher un intervalle de valeurs dans le champ numérique. Utile dans des champs tels que: ID Hadith (idHadith), hadithNum par exemple : hadithNum: [1 à 5], pour récupérer les cinq premiers hadiths de chaque collection.

3.2. Fonctionnalités supplémentaires:

– **Correction orthographique:** fournit des alternatives aux mots clés lorsqu'ils sont mal orthographiés ou différents dans l'AL-Hadith. Par exemple, une proposition pour أبراهام (Abraham): إبراهيم (Ibrahim) comme mentionné dans l'hadith.

– **Sortie en temps réel :** utilisée pour éviter le temps d'attente de l'utilisateur et afficher les résultats directement lorsque vous les récupérez.

– **Scoring:** attribuez à chaque résultat un score calculé qui représente la pertinence de ce résultat pour la requête de recherche. Ce score sera utilisé pour le tri et / ou le filtrage.

– **Agrégation des résultats** utilisée pour offrir une meilleure exploration des résultats, car l'utilisation de "chapter" comme unité d'affichage n'est pas toujours le meilleur choix pour les utilisateurs. Il est parfois plus utile d'agrégé ces "chapters" sur des "books", ou tout autre critère. Les critères qui peuvent être utilisés pour l'agrégation sont: par book (الكتاب), par chapter (الباب), par narrator (راوي الحديث), par author(مخرج الحديث), par collection (مصدر الحديث), et par fiabilité (حكم الحديث).

3.3. Aspects linguistiques :

– **Dérivation multi niveau** : cette fonctionnalité est utile pour localiser tout ou partie des formes de dérivation de mot. Les dérivations de mots arabes peuvent être divisées en quatre niveaux : **le mot exact** (فأسقيناكموه "Nous te le donnons donc à boire"), **Lemma** (أسقينا "Nous donnons à boire"), **stemm** (أسقى " J'ai donné à boire"), **Racine** (سقي). Par exemple: (mot: أسقينا, niveau: lemme) à trouver وَأَسْقَيْنَاكُمْ ,لَأَسْقَيْنَاهُمْ , فَأَسْقَيْنَاكُمُوهُ .

– **Dérivés spécifiques** : il s'agit de la spécification de la fonctionnalité précédente. Depuis l'arabe est complètement courbé, il a de nombreux dérivés. Par exemple au passé : قال (dire) trouver قالت (* elle* dit), قال (* il * dit), قالوا (* ils *, dit), قلن (* elles *, dit), etc.

– **intégré la requête aux propriétés de mot**: fournit un moyen intelligent de traiter les familles de mots en filtrant à l'utiliser d'un ensemble de propriétés telles que: racine, lemme, genre, type, humeur, formule du verbe, genre, personne, number, voix, etc. Par exemple : (root: ملك, type : nom, number: singulier) pour récupérer toute la famille root (ملك) qui est sous forme nom et format singulier.

3.4. Fonctions de recherche dans al-hadith :

- **Développer la recherche :** Il est préférable que le moteur de recherche inclue même les types de fiabilités et d'explications et ajoute des informations externes aux al-hadiths par exemple lorsque cela est mentionné dans d'autres collection (source) d'al-hadith**.**
- **Recherche avec des options d'al-hadith structurée :** al-hadith est connu pour sa structure interne structurée. il est divisé en collection (source) et chaque collection divisé en book (livres) et book divisé en chapter et en chapter divisé en al-hadith. Cette option est l'une des options les plus importantes. Elle indique à l'utilisateur comment réduire le champ de recherche à sa guise.
- **Fournir des requêtes statistiques :** Fournir des requêtes statistiques à ceux qui veulent que les chercheurs en charia sachent combien de fois.Et apportez des réponses à ces questions:

  Combien de fois le mot "محمد" est-il apparu dans la collection "صحيح البخاري".

4. Modélisation :

4.1. Diagramme de classe (UML) :

Les éléments du texte Hadith ont été extraits et structurés dans une base de données relationnelle. Le modèle conceptuel de notre base de données *Hadith* est présenté dans la figure suivant : à un book au moins.

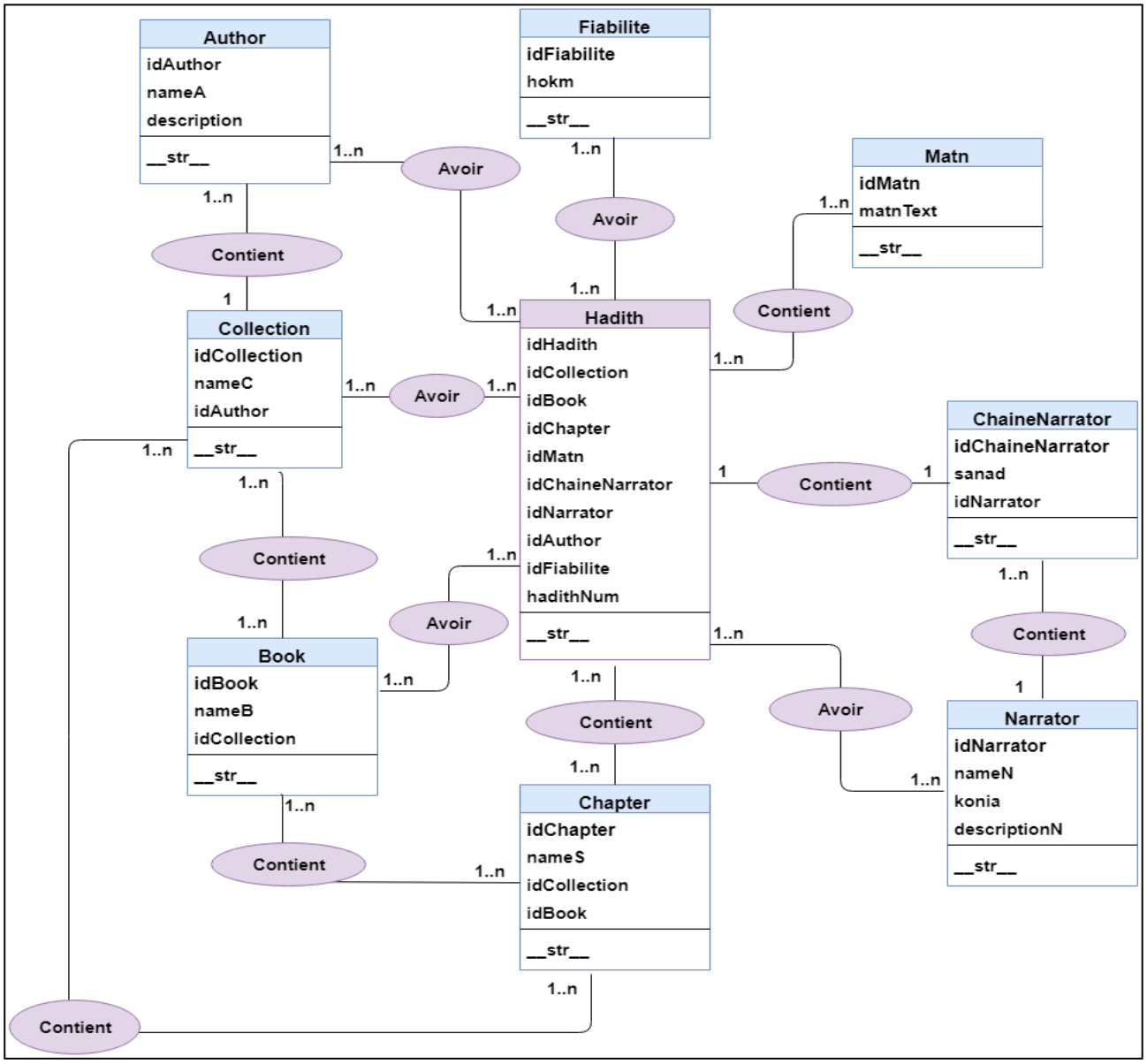


Figure 4.2 : Diagramme de classe d'al-hadith

On cite ci-dessous les informations qui découlent des principes de la modélisation (entité/relation) entre les tables de notre base de données :

- **Collection et Author :**
  - Chaque author a obligatoirement une collection et elle peut en contient plusieurs.
  - Chaque collection est associée obligatoirement à un author et une seule.
- **Collection et Book :**
  - Chaque collection contient obligatoirement un book mais il peut en contient plusieurs.
  - Chaque book est associé obligatoirement à une collection au moins.
- **Collection et Chapter :**
  - Chaque collection contient obligatoirement un chapter mais il peut en contient plusieurs.

- Chaque chapter est associé obligatoirement à une collection au moins.

– **Book et Chapter :**

- Chaque book contient obligatoirement une chapter et elle peut en contient plusieurs.
- Chaque chapter est associé obligatoirement à un book au moins.

– **ChaineNarrator et Narrator:**

- Chaque narrator a obligatoirement une chaineNarrator et elle peut en contient plusieurs.
- Chaque chaineNarrator est associé obligatoirement à une narrator et une seule.

– **Collection et Hadith :**

- Chaque collection contient obligatoirement un hadith mais il peut en contient plusieurs.
- Chaque hadith est associé obligatoirement à une collection au moins.

– **Hadith et Book :**

- Chaque book avoir obligatoirement un hadith mais il peut en avoir plusieurs.
- Chaque hadith est associé obligatoirement à un book au moins.

– **Hadith et Chapter :**

- Chaque chapter contient obligatoirement un hadith, mais il peut contenir plusieurs.
- Chaque hadith est associé obligatoirement à un chapter au moins.

– **Hadith et ChaineNarrator :**

- Chaque chaineNarrator a obligatoirement un hadith et seulement une.
- Chaque hadith est associé obligatoirement à une chaineNarrator et une seule.

– **Hadith et Narrator :**

- Chaque narrator avoir obligatoirement un hadith mais il peut en avoir plusieurs.
- Chaque hadith est associé obligatoirement à un narrator au moins.

– **Hadith et Matn :**

- Chaque hadith contient obligatoirement un matn et seulement une.
- Chaque matn est associé obligatoirement à un hadith au moins.

– **Hadith et Fiabilite :**

- Chaque hadith avoir obligatoirement une fiabilité mais il peut en avoir plusieurs.
- Chaque fiabilité est associée obligatoirement à un hadith au moins.

– **Hadith et Author :**

- Chaque hadith avoir obligatoirement un author mais il peut en avoir plusieurs.
- Chaque author est associée obligatoirement à un hadith au moins.

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

Collection		
Nom du champ	**Type**	**Description**
idCollection	INTEGER	Identificateur de la collection
nameC	TEXT	Name de la collection
idAuthor	INTEGER	Identificateur d'author
Book		
idBook	INTEGER	Identificateur du book
nameB	TEXT	Name du book
idCollection	INTEGER	Identificateur de la collection
Chapter		
idChapter	INTEGER	Identificateur du chapter
nameS	TEXT	Name du chapter
idCollection	INTEGER	Identificateur de la collection
idBook	INTEGER	Identificateur du book
ChaineNarrator		
idChaineNarrator	INTEGER	Identificateur du chaineNarrator
Sanad	TEXT	Description du sanad
IdNarrator	INTEGER	Identificateur du narrator
Matn		
idMatn	INTEGER	Identificateur du matn

matnText	TEXT	Contenant du matn
Narrator		
idNarrator	INTEGER	Identificateur du narrator
nameN	TEXT	Name du narrator
Konia	TEXT	Konia du narrator
descriptionN	TEXT	Plus informations du narrator
Fiabilité		
idFiabilite	INTEGER	Identificateur de la fiabilité
Hokm	TEXT	Fiabilité du hadith
Author		
idAuthor	INTEGER	Identificateur d'author
nameA	TEXT	Name du mokhrij al-hadith
descriptionA	TEXT	Plus informations d'author
Hadith		
idHadith	INTEGER	Identificateur du hadith
hadithNum	INTEGER	Numéro du hadith dans la collection
idAuthor	INTEGER	Identificateur d'author
idCollection	INTEGER	Identificateur de la collection
idBook	INTEGER	Identificateur du book
idChapter	INTEGER	Identificateur du chapter

idChaineNarrator	INTEGER	Identificateur du chaineNarrator
idMatn	INTEGER	Identificateur du matn
idNarrator	INTEGER	Identificateur du narrator
idFiabilite	INTEGER	Identificateur de la fiabilité

Tableau 4.1 : Description des tables de la base Hadith

5. Méthode de recherche :

Le travail consiste à développer un moteur de recherche basé sur Elasticsearch pour atteindre notre objectif, nous nous sommes basés sur le comportement général des systèmes de recherche.

Le chercheur est l'élément fourni pour effectuer la recherche et est elasticsearch. Obtient les requêtes des utilisateurs, et récupère dans un index inversé d'al-hadith pour obtenir les identifiants de documents d'al-hadith identiques, puis utilise ces identifiants de documents pour récupérer toutes les informations des al-hadiths correspondantes, telles que le texte arabe (Matn : المتن) et le nom du narrator (rawi : راوي الحديث). Des informations complètes seront envoyées aux interfaces en tant que résultats. Le comportement du chercheur est décrit dans cette figure :

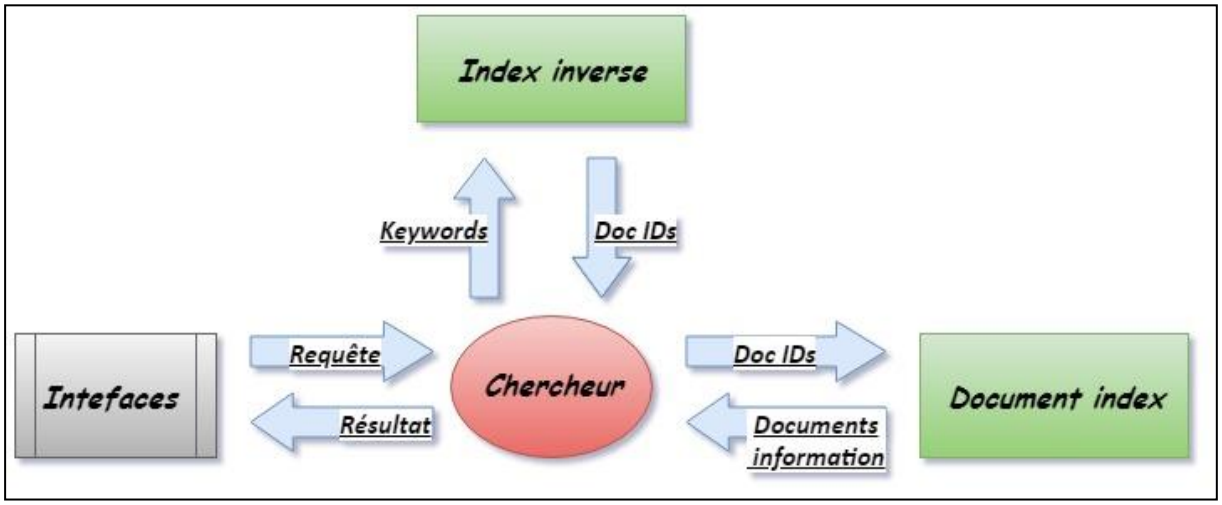


Figure 4.3 : Prototype de base de recherche

6. Architecture de notre système :

Systèmes de récupération d'informations qui identifient des mots en fonction des critères de correspondance entre les mots des requêtes de l'utilisateur et ceux des documents.

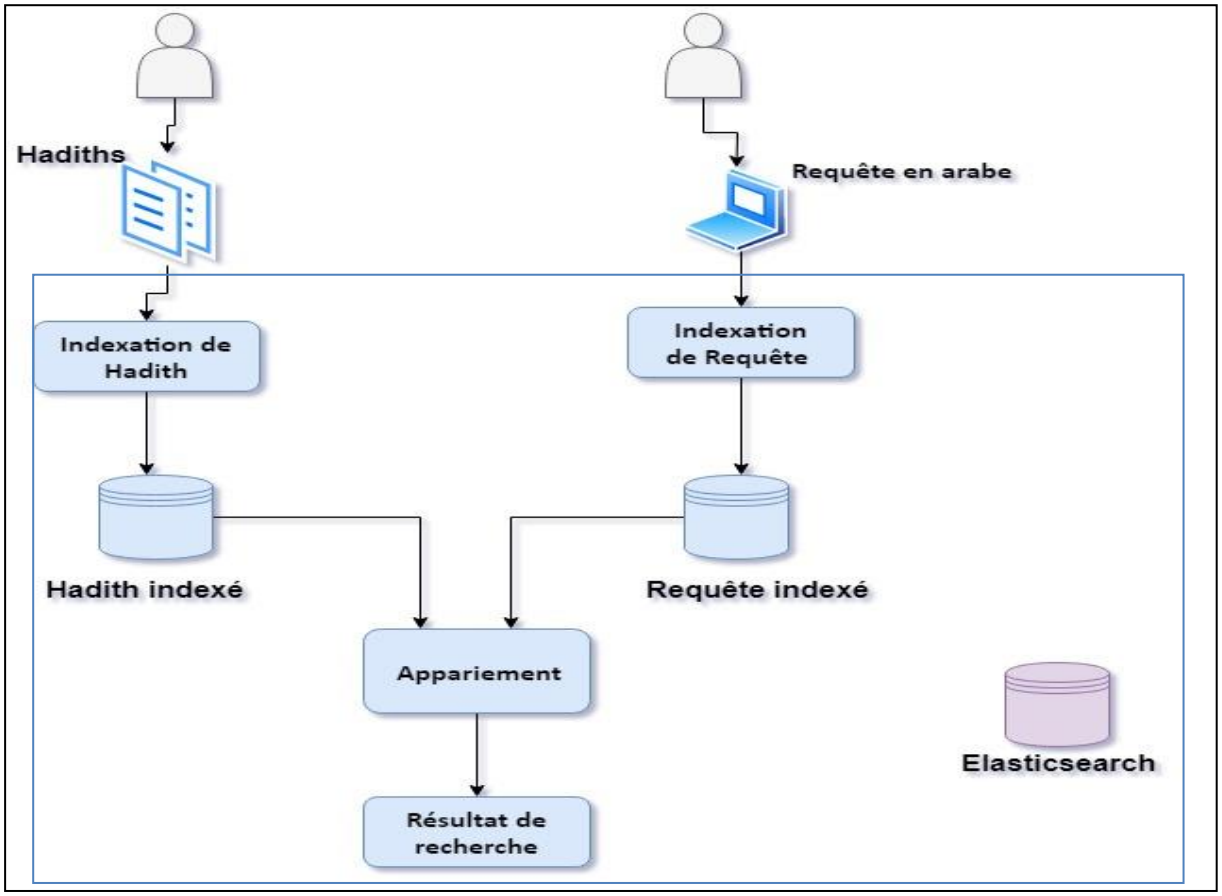


Figure 4.4 : L'architecture de notre système**.**

Notre système de RI se compose de trois modules importants :

– Le module d'indexation de document.

- Le module d'indexation de la requête.

- Le module d'appariement.

Commence par l'utilisateur est charger les documents, Ensuite elasticsearch indexe ces documents, et crée une base indexée (Hadith indexé). La requête de l'utilisateur est aussi indexée.

Ces étapes peuvent être considérées comme une préparation à la recherche. Son but est d'extraire les termes corrects avec une représentation détaillée du contenu sémantique de la requête ou du document. L'indexation est une étape fondamentale dans la conception d'un système de recherche d'information, cette étape permet de passer d'une description brute d'un document, d'un concept ou d'une requête vers une description structurée qui se compose d'une liste de termes significatifs

pondérés par des poids donnant l'importance à ces derniers l'ensemble des indexes rassemblés dans un dictionnaire.

Le troisième model est l'appariement entre les termes de la requête d'un utilisateur et les documents (الاحاديث), s'effectue au niveau de l'appariement "document - requête", cette étape sert à renvoyer une liste de documents ordonnées selon le degré de pertinence, ce dernier est calculé à partir d'une fonction notée Score (Q, D), où **Q** est une requête et **D** un document. L'expression de la fonction d'appariement est tributaire du modèle de RI choisi (eleasticsearch).

6.1. Traitement d'al-Hadith :

Le traitement de texte est généralement basé sur deux phases principales. Le premier est le tokenization (extraire de tokens). Le second est le traitement de ces tokens, il inclut la normalisation des caractères, le filtrage des mots vides, et trouver les stemms. Nous basons sur les mêmes phases mais nous les avons personnalisées comme définie dans la figure suivant :

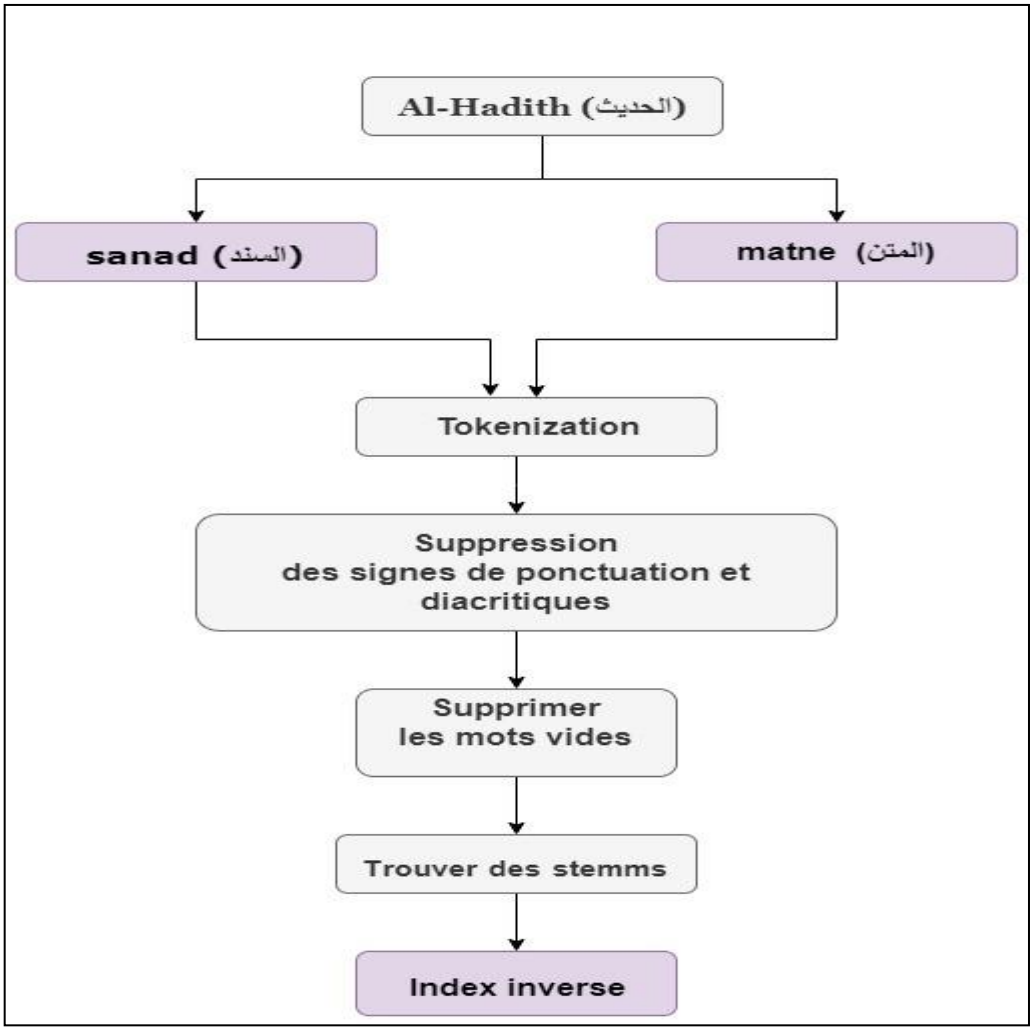


Figure 4.5 : Traitement de texte d'al-hadith

6.1.1. Prétraitement :

– **Tokenization** : vise à diviser al-hadith en tokens (mots), le token al-hadith est facilement résolu car chaque token peut être spécifié en tant que chaîne de caractères entre des espaces**.**

– **Suppression des signes de ponctuation et diacritiques :** il est importantes de supprimer la ponctuation et les signes diacritiques, car ces balises sont courantes dans al-hadith et n'ont aucun effet sur la définition de la classe d'al-hadith.

– **Supprimer les mots vides :** les mots vides sont des mots qui se trouvent dans al-hadith et qui n'ont aucune signification dans le système proposé, il est composé de pronoms et de prépositions arabes. Après avoir supprimé les mots vides d'al-hadith, les mots restants (termes) sont considérés comme des fonctionnalités.

– **Trouver des stemms :** l'extraction de stem final est considérée comme une extraction de stem légère reposant sur la suppression de certains préfixes ou suffixes du mot pour lier à ses racines, nous avons utilisé un algorithme radical (stem). L'extraction du stem filtrée a eu pour résultat d'éliminer les stemms incorrects. Les stemms résultants seront utilisés dans le processus de développement de la requête.

6.1.2. Index inversé :

La structure d'index inversé est à la base de la plupart des systèmes de recherche d'information. Dans cette structure, chaque al-hadith (document) peut être représenté par une liste de mots clés qui décrivent le contenu du document pour la recherche. Une recherche rapide peut être réalisée si on inverse ces mots clé. Des mots clés sont stockés par exemple par ordre alphabétique dans l'index. Une liste de pointeurs aux documents de qualification dans les "fichiers postant " est maintenue. Cette méthode est utilisée dans la plupart des systèmes d'informatiques actuels. <Doc Id, t_f> t_f : la fréquence de mot dans cette doc ID.

Figure 4.6 : Structure d'index inversé

5.1.1. Exemple de traitement de document (texte al-hadith) :

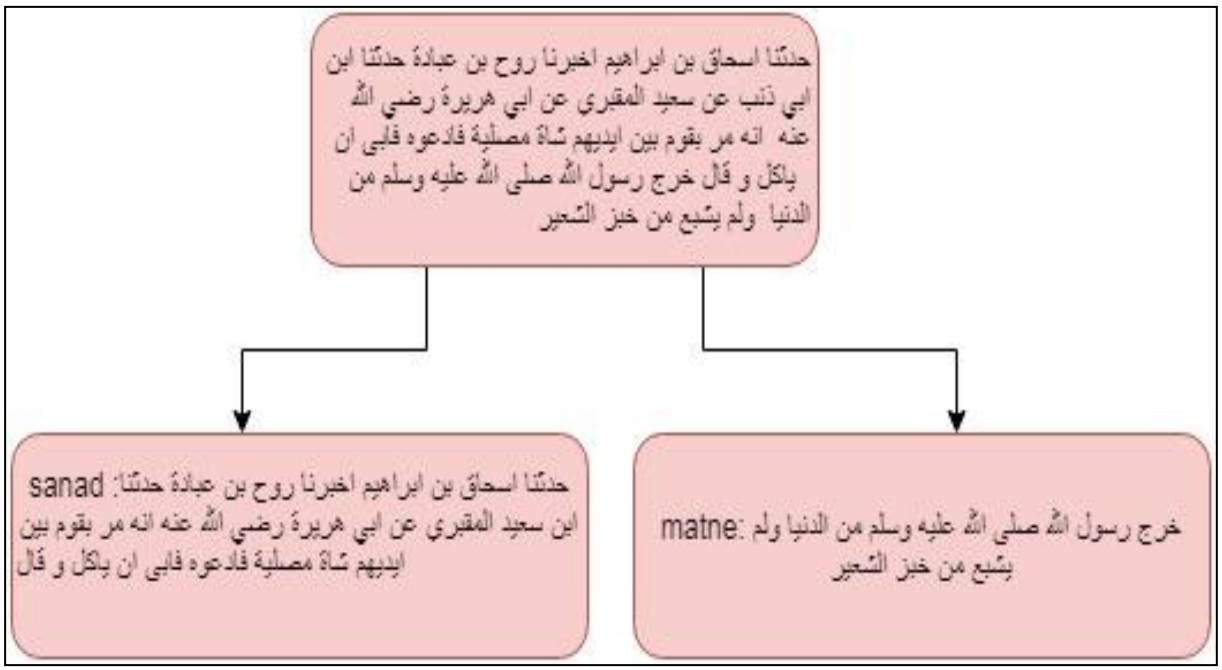


Figure 4.7 : Exemple d'al-hadith.

Étape	Résultat de l'étape
Tokenization	{ "حدثني", "إسحاق", "بن", "إبراهيم", "أخبرنا", "روح", "بن", "عبادة", "حدثنا", "ابن", "أبي", "ذنب", "عن", "سعيد", "المقبري", "عن", "أبي", "هريرة", "رضي", "الله", "عنه", "أنه", "مر", " بقوم ", "بين ", أيديهم ", شاة", "مصلِيَّة ", ف ", "دعوه ", ف", "أبى ", أن " , " يأكل ", و", "قال ", : ", خرج ", رسول ", الله " , " صلى ", الله ", عليه ", و", "سلم ", من ", الدنيا ", ولم" ", يشبع ", من ", خبز " ", الشعير "}
Suppression des signes de ponctuation et diacritiques	{حدثني, إسحاق, بن, إبراهيم, أخبرنا ,روح, بن, عبادة, حدثنا,ابن, أبي, ذنب, عن, سعيد, المقبري , عن,أبي ,هريرة, رضي, الله ,عنه,أنه,مر , بقوم , بين ,أيديهم ,شاةَ,مصلِيَّة, فدعوه , فأبى , أن , يأكل , وقال, خرج , رسول , الله , صلى, الله, عليه, و, سلم , من, الدنيا , ولم, يشبع , من , خبز , الشعير }
Supprimer les mots vides	{حدثني, إسحاق, بن, إبراهيم, أخبرنا , روح , عبادة ,حدثنا , ابن , أبي , ذنب, سعيد , المقبري , أبي ,هريرة , رضي , الله , مر , بقوم, ايديهم, شاة ،, مصلية, فدعوه, فأبى , يأكل , خرج ,الله , صلى , عليه, وسلم , الدنيا , يشبع , خبز ,الشعير}
Stemms	{ إسحاق , إبراهيم, روح ,عبادة, ذنب, سعيد, المقبري, هريرة , حدث ,اخبر ,شاة, أيدي,صلى, خرج, اكل ،شبع, دنيا, شعير}

6.2. Traitement des requêtes:

- Pour chaque terme (T) de la requête :
  - Obtenir la liste d'al-hadiths contenant (T), en utilisant l'index inversé
  - Pour chaque al-hadith (H) de cette liste:
    - Mettez à jour le score de (H) en fonction des poids de (T) dans l'al-hadith et dans la requête :

$$\text{Score (H)} = \text{(poids (T, H) * poids (T, R)}$$

    - Mettre à jour la somme des carrés de poids utilisés pour le décompte du score pour (H) et pour (R) (c'est-à-dire utilisés pour la normalisation des scores, ils représentent la norme des vecteurs d'al-hadiths et de la requête),
- Normaliser les scores de chaque al-hadith.
- Commandez l'al-hadiths selon les scores normalisés.

6.2.1. Exemple de requête :

Requête: تسموا باسمي ولا تكنوا بكنيتي

Un exemple de requête est représenté par le besoin d'authentification d'al−hadith n ° 3974 de la collection «صحيح مسلم» qui concerne les noms (الكني) du Prophète (محمد صلى الله عليه وسلم). Nous avons extrait deux racines du texte du Matne sont: «سما » pour « تسموا ,باسمي » et « كنا » pour « تكنوا ,بكنيتي »,, ces racines apparaissent dans la requête deux fois pour chacune.

Par conséquence, nous avons calculé le poids selon la fonction de pondération (TFI*DF) qu'effectué sur un ensemble de 54 hadiths, la racine « سما » apparaît 29 fois dans 19 hadiths, et que la racine «كنا» apparaît 12 fois dans 6 hadiths. La racine "كنا" est un poids de 7,41, ces qui s'explique par le fait que la mesure (TFI*DF) fournir un pouvoir discriminatoire important aux termes rares de la base des al-hadiths.

Où : TF*IDF= TF * $\log_2(\frac{N}{N_i})$

- TF : Nombre de terme dans le document (hadith)
- N : Nombre de document de la collection.
- N_i : Nombre de document qui contient le terme i.

7. Conclusion :

Ce chapitre a été consacré à la description conceptuelle de notre approche pour l'indexation, la classification des fonctionnalités de recherche et le référencement d'al-hadith, avec la suggestion des corrections pour des résultats améliorations, aussi présentée la base relationnelle d'al-hadith et l'architecture de système est expliqué. Un modèle de recherche est proposé pour répondre à n'importe quelle requête pour l'utilisateur.

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

Chapitre 05 : Implimentation

1. Introduction :

Notre objectif de mise en œuvre est de fournir une API open. L'API doit être bien évolutive pour maximiser les fonctionnalités de recherche dont nous avons discuté dans le chapitre précédent. Dans ce chapitre, nous présentons les outils utilisés pour développer et implémenter notre système, et nous expliquons pourquoi nous avons décidé d'utiliser open source.

2. Ressources utilisé :

2.1. Python :

Python est un langage de programmation dynamique est utilisé dans une grande variété de domaines d'application. Python est souvent comparé à Tcl, Perl, Ruby, Scheme ou Java.
Nous avons choisi le langage Python car il présente certaines de ses principales caractéristiques:

- Python est open source, ce qui le rend gratuit à utiliser et à distribuer.
- Il est indépendant des systèmes et fonctionne sur différents systèmes: Windows, Linux , Mac, Amiga, et sans changer le code.
- Il prend en charge Unicode dans sa nature et donc la langue arabe.

2.2. Elasticsearch :

Nous avons choisi elasticsearch car :

- Elasticsearch est un moteur de recherche open source, utilise API.
- La plupart des BDD sont inadéquates à extraire des données exploitables, bien sûr elles peuvent filtrer par date ou par valeurs exactes mais ne peuvent pas faire une recherche en plein texte, et trier des documents par pertinence. Et surtout, elles ne le font pas en temps réel, contraire elasticsearch qui fait tout ça.

3. Pourquoi Open Source?

L'une des différences fondamentales entre les logiciels open source et les logiciels propriétaires est que le code source des logiciels open source doit être mis gratuitement à disposition avec le logiciel. Tout le monde devrait pouvoir télécharger le code source, le visualiser et le modifier comme bon lui semble.

Il existe un certain nombre d'avantages qui nous conduisent à l'open source, les points suivants examinent les plus importants d'entre eux :

- Correction collaborative de bogues: comme un grand nombre d'utilisateurs peuvent accéder au code et le modifier, les bogues ont tendance à être plus visibles et plus rapidement corrigés.

– Détection rapide des vulnérabilités de sécurité: l'accès au code source facilite et accélère la détection desvulnérabilités de sécurité dans les logiciels, que vous cherchiez à les corriger ou à les exploiter, cela suggérer que le code open source est moins sécurisé que le code source fermé.

– Traduction et localisation: grâce à l'accès au code source, traduction facile de la langue de l'interface du programme.

– Arrêtez le développement: avec les logiciels open source, le risque d'indisponibilité est considérablement réduit, car il peut être capturé et développé le code source par toute personne intéressée par la survie du produit.

4. Implémentation des fonctionnalités de recherche :

Nous avons inclus de nombreuses fonctionnalités de recherche dans chapitre précédent (voir 4.3). Bien que nous n'ayons pas mis en œuvre la liste complète. Nous sommes .Nous mentionnons l'état de mise en œuvre dans le tableau suivant :

Classification	**Fonctionnalité**	**la mise en œuvre**
Fonctionnalités Recherche avance	Recherche de relation logique	Partiellement
	Recherche de champs	oui
	Recherche de phrase	oui
	Recherche d'Intervalle	non
Fonctionnalités supplémentaire	Correction orthographe	non
	Sortie en temps reel	oui
	Scoring	oui
	Agrégation des résultats	non
Aspects linguistiques	Dérivation multi niveau	oui
	Dérivés spécifiques	oui
	Intégré la requéte aux propriétés de mots	Oui

Fonctions de recherche dans al-hadith	Développer la recherche	Non
	Recherche avec des options d'al-hadith structurée	Oui
	Fournir des requêtes statistiques	Oui

Table 5.1 : Implémentation des fonctionnalités de recherche.

4.1. Présentation de l'application :

Le but de cette application est de fournir une interface pleine de fonctionnalités dans laquelle nous utilisons elasticsearch, qui est un moteur de recherche open source qui permet de tri les documents par pertinence en temps réel et de rechercher texte intégral.

Lors de la recherche du hadith que nous voulons donnée par un mot ou d'une phrase, cette application nous donne: le hadith et toutes les informations d'al-hadith .et cette application permet aussi de rechercher des options d'al-hadith structurée (collection, book, chapter, narrator).

4.1.1. Interface principale :

Voila interface principale d'application :

Figure 5.1 : Interface principale

4.1.2. Rechercher par terme:

Par exemple pour rechercher par terme : الزكاة voila le résultat :

Figure 5.2 : Rechercher par terme

4.1.3. Rechercher par phrase :

– **Rechercher par sanad :**

Figure 5.3 : Rechercher par sanad.

– **Recherche par matn :**

Figure 5.4 : Rechercher par matn

4.1.4. Rechercher par des options d'al-hadith structurée :

– **Rechercher par une collection :**

Figure 5.5 : Rechercher par collection

- **Rechercher par un Books :**

Figure 5.6 : Rechercher par book

- **Rechercher par un Chapter :**

Figure 5.7 : Rechercher par chapter

– **Rechercher par un narrator :**

Figure 5.8 : Recherche par narrator

5. Analyser et restaurer de données avec Kibana:

Elasticsearch peut etre indexer de nombreuses données, et fonctionne par index, et contenu.Il y a de nombreuses extensions sont disponible, dont un des plus connu est Kibana qui est utilisés pour analyser et agréger et restaurer les données en la visualisation de différentes manières.

– Par exemple premierement nous on obtenir tous les documents d'index al-hadiths triés par dates décroissantes :

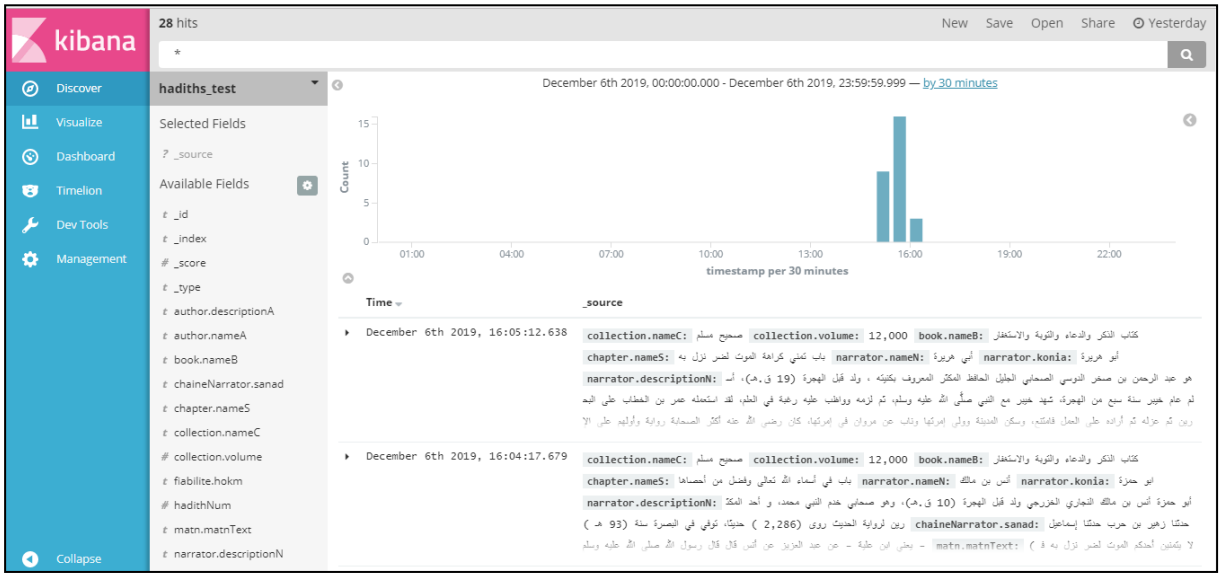


Figure 5.9 : Le fond de discover tous les documents d'index al-hadiths

– Ensuite nous utilisons d'une visualisation de graphique à secteurs (Pie chart), pour afficher les les 5 premier narrateurs ont raconté la plupart des hadiths :

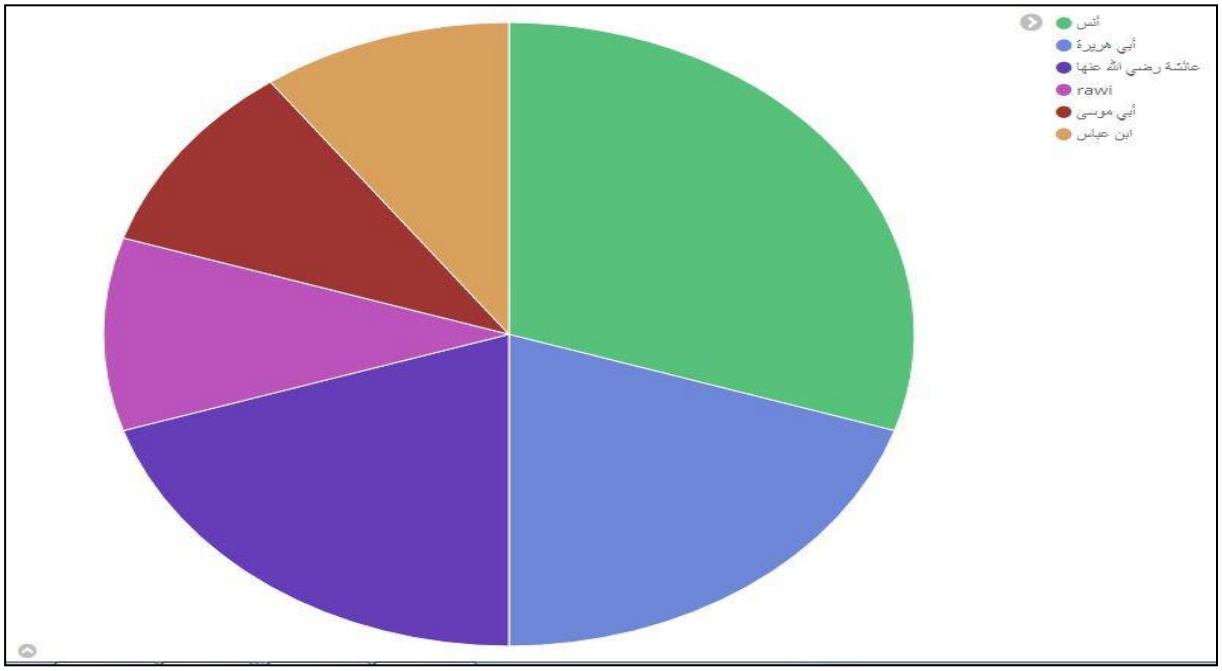


Figure 5.10 : Les 5 premier narrateurs ont raconté la plupart des hadiths

– Et enfin, nous voulons que les collections d'al-hadith représentent un certain nombre de documents, puis divisez chaque collection de collections en books et chaque book en chapter Voici le résultat :

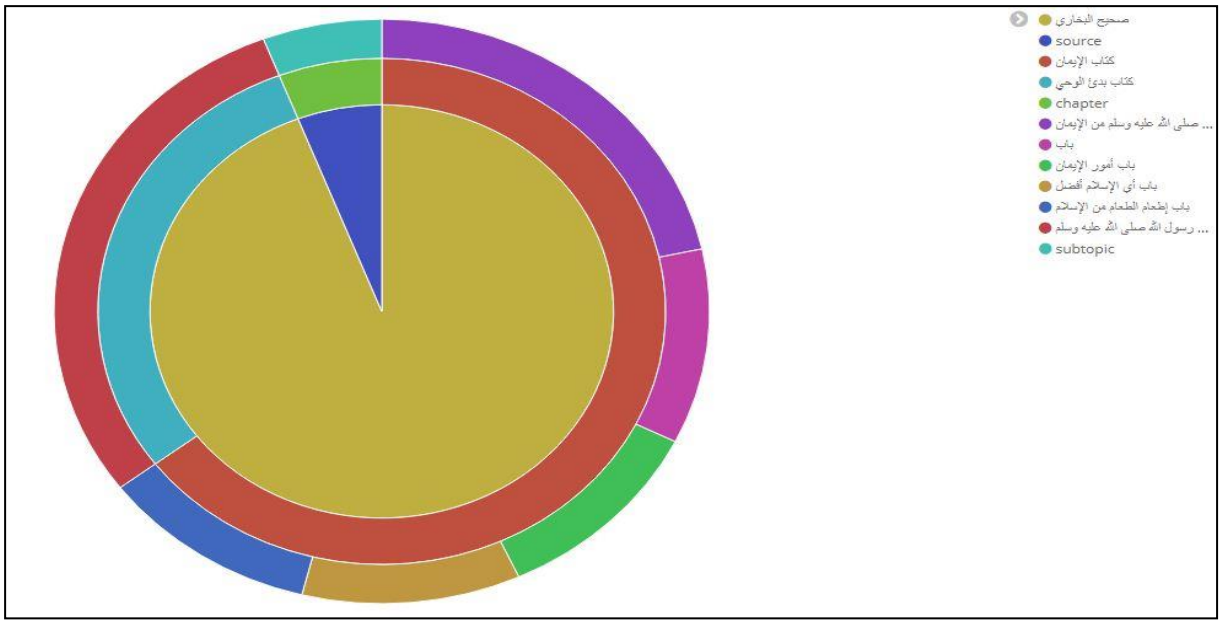


Figure 5.11 : Écran de visualisation des genres classifié d'al-hadith

6. Traitement d'un exemple des requêtes:

- **Requête (rechercher par phrase)**: تسموا باسمي ولا تكنوا بكنيتي

  - **Calculer TF*IDF :**

TF*IDF ('كنا',3974)=$TF_{3974,\text{كنا}}$*$IDF_{3974,\text{كنا}}$

=2*(($\log_2 \frac{54}{6}$) +1)

= 7,41

TF*IDF ('سما',3974)=$TF_{3974,\text{سما}}$*$IDF_{3974,\text{سما}}$

=2*(($\log_2 \frac{54}{19}$) +1)

=3 ,71

- **Les résultats de la pondération sont représentés dans le tableau suivant:**

Terme de la requête	Poids TF* IDF
كنا	7,41
سما	3,71

- **Les hadiths donnés sont les suivants :**
  - Hadiths N=3976, 3977, 3978, 3979 du «Sahih Musslem»:
  - Hadith N=3976 {فإنما أنا قاسم أقسم بينكم تسموا باسمي ولا تكنوا بكنيتي}
  - Hadith N=3977 {فإنما بعثت قاسما يقسم بينكم سموا باسمي ولا تكنوا بكنيتي}
  - Hadith N=3978 {فإنما أنا أبو القاسم أقسم بينكم تسموا باسمي ولا تكنوا بكنيتي}
  - Hadith N=3979 {تسموا باسمي ولا تكنوا بكنيتي}

Terme de la requête	**Poids pour le hadith 3976**	**Poids pour le hadith 3977**	**Poids pour le hadith 3978**	**Poids pour le hadith 3979**
كنا	7 .41	7.41	7.41	7.41
سما	12.78	3.71	3.71	3.71
بعث	0.0	17.25	0.0	0/0
قسم	10.12	10.12	5.06	0.0

Table 5.2 : Index termes des hadiths pertinents pour la requête.

Terme de la requête	Score pour le hadith 3976	Score pour le hadith 3977	Score pour le hadith 3978	Score pour le hadith 3979
كنا	54.90	54.90	54.90	54.90
سما	48.15	13.76	13.76	13.76
Cos (d ,r)	0 ,68	0.38	0.85	1.00

Table 5.3 : Score de hadiths

$$cosine(d,r) = \frac{\sum_{w \in d \cap r} TFIDF_{w,d} \bullet TFIDF_{w,r}}{\sqrt{\left(\sum_{w \in d} TFIDF_{w,d}^{\ 2}\right) \bullet \left(\sum_{w \in r} TFIDF_{w,r}^{\ 2}\right)}}$$

$$\text{Score} = \frac{((7.41*7.41)+(12.98*3.71))}{\sqrt{((7.41)^2+(3.71)^2)*((7.41)^2+(12.98)^2))}}$$

$$= 0\ .68$$

- Donc : le resultat des hadiths sonts :

1- Hadith N=3979 {تسموا باسمي ولا تكنوا بكنيتي}

2- Hadith N=3978 {فإنما أنا أبو القاسم أقسم بينكم تسموا باسمي ولا تكنوا بكنيتي}

3- Hadith N=3976 {فإنما أنا قاسم أقسم بينكم تسموا باسمي ولا تكنوا بكنيتي}

4- Hadith N=3977 {فإنما بعثت قاسما يقسم بينكم سموا باسمي ولا تكنوا بكنيتي}

7. Conclusion :

Le chapitre explique pourquoi nous avons choisi l'approche open source. Nous avons parlé des améliorations que nous avons apportées. Nous avons accompli beaucoup, mais malheureusement, il y a encore des améliorations à apporter et de nombreux problèmes à résoudre.

Conclusion générale

Notre proposition est un développement d'un moteur de recherche open source dans le Hadith basé sur elasticsearch afin d'offrir ou utilisateurs une interface pleine de fonctionnalité est aux développeurs une API riche. Mais pour atteindre cet objectif, nous avons d'abord du répertorier et classer toutes les fonctionnalités de recherche possibles et utiles. Ensuite, nous devions concepteur le système de recherche pour réaliser chaque fonction de recherche.

Nous avons commencé avec trois chapitres sur l'état de l'art. Le premier chapitre concerne étudier les aspects informationnels d'al-hadiths, le second chapitre concerne les moteurs de recherche et leur fonctionnement, et le troisième chapitre concerne les fonctionnalités ElasticSearch.

Pour illustrer notre proposition, nous avons classé les fonctions de recherche dans al-hadiths qui utiles. Ces fonctions nous aideront à récupérer un détail répondant aux exigences d'al-hadith., en s'appuyant sur elasticsearch qui permet la recherche en texte intégral, le tri des documents par pertinence dans temps réel.

Pour la conception, nous avons examiné une explication de ce qu'il faut faire, puis nous avons proposé une classification des fonctionnalités de recherche et plusieurs améliorations.

La première amélioration concerne la réalisation d'un moteur de recherche personnalisé complet offrant la plupart des fonctionnalités mentionnées dans le chapitre de conception.

La deuxième amélioration consiste à proposer un traitement de texte spécifier qui convient aux caractéristiques du texte hadith

Troisième amélioration, nous avons considéré le mot (recherche par champ) comme une unité de recherche et matn une unité principale. L'objectif est d'obtenir un ensemble de fonctionnalités de recherche nécessitant le traitement de combinaisons de mots. La recherche directe ou indirecte d'un mot dans son ensemble aide à de nombreuses fonctionnalités:

- Recherche aux propriétés de mot.
- Recherche par synonymes.
- Dérivés multi niveaux.
- Dérivés spécifiques.

Nous avons examiné de nombreuses fonctions de recherche mentionnées précédemment. Malheureusement, il y a plus d'améliorations à apporter et de nombreux problèmes à résoudre.

Nous leur avons laissé des perspectives:

- Réaliser un système pour séparer sanad du Matn.
- Mise en œuvre du système de proposition le plus approprié.
- Suivez la visualisation complète de toutes les fonctions de recherche mentionnées au chapitre 4 du titre 4.

Bibliographie

[البواب 2009] مروان البواب.محركات البحث في النصوص العربية.مجمع اللغة العربية.دمشق.2009.

[الخطيب2010] : منار الخطيب. تصنيف الحديث الشريف باستخدام خوارزمية التنقيب عن البيانات. مؤتمر أوروبا والبحر الأبيض المتوسط والشرق الأوسط حول نظم المعلومات ، أبو ظبي ، الإمارات العربية المتحدة.2010.

[عزمي2017] :عزمي ، العوفيلي. طريقة جديدة لتمرير الحكم تلقائيًا على الحديث. المؤتمر الدولي الخامس حول معالجة اللغة العربية ، وجدة ، المغرب. 2017.

[طحان1985]: طحان م. مقدمة في علوم الحديث. طبعة المعارف ، الرياض ، المملكة العربية السعودية. 1985.

[الحربي2008] : الحربي ، الثبيتي ، خورشيد ، الراجح. تصنيف النص العربي التلقائي. مدينة الملك عبد العزيز للعلوم والتقنية ، الرياض ، المملكة العربية السعودية. 2008.

[ذياب1434] : ذياب الغامدي .توريق المنة لحفاظ الأسانيد و السنة .الطائف المانوس .1434.

[نور الدين 1997] : نور الدين عنتر .منهج النقد في علوم الحديث .دار الفكر .1997.

[Abar2009] : Ali Abar, Mounir Boughanem. La capitalisation du profil des experts du ceneap sur la base de la classification des etudes. Thesis, Institut National de Formation en Informatique, Algiers. 2009.

[Abbes2008] : International Conference on Web and Information Technologies.Sidi Bel Abbes, Algeria.2008.

[Abulhajjaj2009]: M.B. Abulhajjaj. Semantic Web - the next revolution.tunis.2009.

[Allab2008] : Kamel Allab, Abdelkrim Doufene. Classification automatique de documents.Oran,Algeria.2008.

[Amrouche2008] : Karima Amrouche. Passage à l'echelle en recherche d'information : Methode d'elagage pour la reduction de l'espace de recherche. Thesis, Institut National de Formation en Informatique, Algiers. 2008.

[Antoine2015] : Antoine Barbare.Stack ELK : logstash et kibana les outils d'Elasticsearch. Packt Publishing Ltd. 2015.

[Azami1978] : Al-Azami MM, Studies in early Hadith literature. American Trust Publication, Indianapolis, 1978.

[Azmi, Badia 2010] : Aqil Azmi, Nawaf Badia, "Tree – Automating the Construction of the Narration Tree of Hadith", IEEE, 2010.

[Batyrzhan2014] : Batyrzhan.M, Kulzhanova BR, Abzhalov SU, Mukhitdinov RS. Significance of the hadith of the Prophet Muhammad in Kazakh proverbs and sayings, data mining and classification: a comparative an Proced Social BehavSci. 2014.

[Bernard2009] : Bernard, E, Griffin, J. Hibernate Search in Action. Manning Publications.2009.

[Brahimi2014] : Sabiha Brahimi, Hamza Kouadri. Amélioration de la précision et du temps de réponse d'un moteur de recherche de texte. Université Kasdi Merbah Ouargla. 2014.

[Brin1998] : Sergey Brin .The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems. Packt Publishing Ltd .1998.

[Chartron1989] : Ghislaine Chartron, Sylvie Dalbin, Marie-Gaelle Monteil, Monique Verillon. Indexation manuelle et indexation automatique.1989.

[Catherine2001] : Catherine Roussey. Une méthode d'indexation sémantique adaptée aux corpus multilingues.L'Institut National des Sciences Appliquées de Lyon.2001.

[Dahak2006] : Fouad Dahak. Indexation des documents semi-structures: Proposition d'une approche basee sur le fichier inverse et le trie. Thesis, Institut National de Formation en Informatique, Algiers. 2006.

[Dahmani2010] : Merouane Dahmani and Assem Chelli and Amar Balla and Taha Zerrouki. Développement d'un moteur de recherche et d'indexation pour les documents coraniques. Engineer thesis, Ecole Nationale Superieure des l'Informatique (ESI) - Algiers, 2010.

[Denoyer2004] : Denoyer. Apprentissage et inference statistique dans les bases de documents structures : Application aux corpus de documents textuels. Doctorate thesis, Université de Paris 6, France.2004.

[Dixit2017] : Dixit B, Kuc Rafal, Rogozinski Marek. Elasticsearch: A Complete Guide. Packt Publishing Ltd, 2017.

[Elhachani1997] : Mabrouka Elhachani. L'indexation automatique.Thesis, Ecole Nationale Superieure des Sciences de l'Information et des Bibliotheques (ENSSIB), 1997.

[Fatt2010] : Fattahizadeh F, Afshari N. An approach to understanding hadith in Wasail al-Shia, Hadith Stud. 2010.

[Grossman2002] : Grossman, Frieder, Goharian. IR Basics of Inverted Index. 2002.

[Hanka1998] : Hanka Hensens. Traitement des fonds de Laboratoires : Indexation Principes.France. Avril 1998.

[Kuc2013] : KUC Rafal, Rogozinski Marek. Elasticsearch server. Packt Publishing Ltd. 2013.

[Lancaster2003]: Lancaster. Indexing and abstracting in theory and practice. France. 2003.

[Mechach2016] : Mechach kheira. Etude de l'impact des methodes de localisation dans les systemes d'information distribuent. Université d'oran.2016.

[Nejjari2007] : Youssef Nejjari. Recherche documentaire sur le web. Université d'oran. 2007.

[Paro2015]: Paro, Alberto. ElasticSearch cookbook. Packt Publishing Ltd. 2015.

[Rafal2013]: Rafal Kuc, Marek Rogozinski.Elasticsearch server.Packt Publishing Ltd. 2013.

[Razvan2016] : Razvan Valafu.article Elastic. Packt Publishing Ltd .2016.

[Robin2016] : Robin Jolliet.Elasticsearch-Indexation et recherche simples. Packt Publishing Ltd .2016.

[Sauvagnat2005] : Karen Sauvagnat. Modele flexible pour la Recherche d'Information dans des corpus de documents semi-structures. Thesis de doctorat, Universite Paul Sabatier, Toulouse, France. 2005.

[Web-RS]: Définition - Que signifie recherche sémantique?. https://www.techopedia.com/definition/23731/semantic-search. viste le site en 16/06/2019