République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option : Réseaux et Systèmes intelligents

Thème

Intégration d'un lemmatiseur arabe dans le cadre d'un système de recherche d'information

Préparés par

Zoulikha BENBLAL et Fatima BELOUAFI

Encadré par

Mr. Mohamed Amine CHERAGUI

Année Universitaire 2014/2015

Remerciements

Nous tenons tout d'abord à remercier puissant « ALLAH » de nous avoir donné la force d'aboutir au terme de ce projet et d'y être arrivé en bonne santé.

Nous ne pouvons pas oublier de présenter notre gratitude à nos parents pour leur patience et les efforts inlassables qu'ils ne cessent de déployer pour nous.

On tien a remercier vivement nos enseignants pour leur présence et leur suivi tout au long de l'année : ***Mr.CHERAGUI Mohamed Amine*** *qui présenté ce sujet, avec ses conseils importantes qui nous ont permis de prendre la bonne direction dans le travail.*

Nos remerciements et notre gratitude vont aux enseignants Faculté des sciences et sciences De la technologie d'Université d'ADRAR.

Merci

Résumé

Le traitement automatique du langage naturel (TALN) est un domaine de l'informatique qui englobe plusieurs thématiques : traducteurs automatiques, générateurs automatiques de résumé, correcteurs orthographiques d'erreurs, la recherche d'information …etc.

La recherche d'information a pour objectif de fournir à un utilisateur un accès facile à l'information qui l'intéresse, cette information étant située dans une masse de documents textuels. A fin d'atteindre cet objectif, il doit représenter, stocker et organiser l'information, Puis fournir à l'utilisateur les éléments correspondant au besoin d'information exprimé par sa requête. La plupart des systèmes de recherche d'information(SRI) utilisent des termes simples pour indexer et retrouver des documents et des modèles tel que : Modèle booléen, modèle vectoriel, modèle probabiliste, modèle de langue,…etc

Notre travail s'inscrit dans le cadre de la recherche d'information dans un domaine spécialité en langue arabe. L'objectif de notre travail est développé un système hybride de quartes (04) modèles et pour améliorer les résultats de notre système en utilisant la lemmatisation des termes.

Mots clés : Traitement automatique du langage naturel (TALN), Recherche d'Information, requête, modèles de recherche, lemmatisation.

مـــلـخـص

المعالجة الآلية للغة الطبيعية مجال معلوماتي يعنى بدراسة العديد من الحقول نذكر منها: الترجمة الآلية، التلخيص ألآلي، المصحح ألآلي و البحث عن المعلومة...الخ.

في دراستنا هذه سنركز عن حقل"البحث عن ألمعلومة"، حيث يمكن تعريفه كمجموعة آليات تعمل على تخزين، تنظيم البحث عن المعلومة واستخراجها من بين كتلة معلوماتية كبيرة حسب احتياج الباحث أو المستعمل بطريقة سهلة و سريعة.

لتبسيط آليات البحث توجد عدة نماذج تستعمل في نظام البحث من أشهرها:

- النموذج الثنائي
- النموذج الشعاعي
- النموذج الاِحتمالي
- النموذج اللغوي

خلال عملنا هذا سنستعمل النماذج الاربعة السابقة الذكر محاولة منا ان ننشئ نموذج هجين يجمع ايجابيات النماذج الأربعة واستبعاد الى حد ما سلبيات كل منها،مع استعمال ما يسمى بـ"المحذر".

الكلمات المفتاحية:

المعالجة الآلية للغة الطبيعية،الترجمة الآلية، التلخيص الآلي المصحح ألآلي،و البحث عن المعلومة، المحذر.

Table des matières

Chapitre 1 : état de l'art (TALN Vs la langue Arabe)

Chapitre 2 : Recherche d'information

Chapitre3: conception et architecture du système IRYSA

Chapitre4 : Résultats et évaluation du système IRYSA

Liste des tableaux

Listes des figures

Introduction générale

Le progrès scientifique actuel nécessite de plus en plus des outils de recherche d'information performants, qui permettent une localisation précise et rapide de l'information désirée. Ces nouveaux besoins ont poussé les chercheures de trouver des méthodes récentes pour rendre cette tâche assez facile et plus efficace.

La recherche d'information est devenue maintenant une branche dans le domaine de l'informatique, il permet à l'utilisateur d'extraire à partir d'une grande base de documents, ceux qui correspondent à ses besoins, d'une manière simple et rapide. L'idée de développer un système de recherche d'information a été évoquée la première fois par les libraires, et cette opération a été automatisée avec l'apparition du premier ordinateur.

Le but de ce travail développe un système de recherche d'information a combiner avec un lemmatiseur en arabe pour obtenir d'un bon résultats.

Ce mémoire est organisé comme suite :

- Le premier chapitre : on a présenté une introduction générale sur la discipline de traitement automatique du langage naturel (TALN), et comme nous travaillions et spécifier la langue arabe donc en va essayons de toucher quelques détails de cette langue.
- Le deuxième chapitre: on parle sur le mécanisme de recherche d'information, et ainsi que le système de recherche d'information et leurs concepts fondamentaux et après en présente les déférentes modèles de recherche d'information les plus connu et en fin les méthodes d'évaluation d'un système de recherche d'information.
- Le troisième chapitre: on présente la conception générale de ce travail avec plus détail et en décrit le système de recherche d'information IRYSA que nous avons le développé.
- Le quatrième chapitre: considère comme un résultat final de ce travail a ce niveau en présente l'interface graphique de système et quelque exemple des testes résultat, ainsi que l'évaluation de notre système.

1. Introduction

Le traitement automatique des langues naturelles(TALN), se situe à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle ; elle concerne la conception et le développement de programmes et techniques informatiques capable de traiter de façon automatique des données exprimées dans une langue naturelle.

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

Les langues naturelles se fondent sur des règles grammaticales, syntaxiques et morphologiques. Le niveau de difficulté et de complexité dépend de la langue elle même. L'arabe est une langue sémitiques très riche et différente des langues occidentales et ceci à plusieurs niveaux. Par exemple, l'écrit de cette langue est sensible au contexte.

2- Bref historique du TALN

Le traitement automatique des langues naturelles est né à la fin des années quarante du siècle dernier, dans un contexte scientifique et politique très précis [1], [2], [3].

- Entre 1951 et 1954 : Zelling Harris publie ses travaux les plus importants de la linguistique (linguistique distributionnaliste) ;
- 1954 : La mise au point du premier traducteur automatique (très rudimentaire) Quelques phrases russes, sélectionnées à l'avance, furent traduites automatiquement en anglais
- 1956 : l'école d'été de Dartmouth, la naissance de l'intelligence artificielle ;
- 1957: N. Chomsky publie ses premiers travaux importants sur la syntaxe des langues naturelles, et sur les relations entre grammaires formelles et grammaires naturelles ;
- 1962: première conférence sur la traduction (Bar-Hillel) rapport AlPAC[1] ;
- 1966: le système ELIZA[2](Weizenbaum66) ;
- 1968: le premier (vrai) système de traduction (Systran , russe ⟶ anglais) ;
- 1971: un système intelligent en mode fermé(SHRDLU[3]) ;

[1] Automatic Language Processing Advisory Committee.
[2] Un programme qui simule un entretien avec un psychiatre.
[3] Un programme qui permet d'interagir avec un robot dans un monde de blocs.

- 1976: le système de traduction METEO mis au point à l'U de M ;
- 80s: système de reconnaissance statistique multi locuteur ;
- 90s : Premiers corpus, approches statistiques apprentissage automatique. Applications utilisent corpus de grande taille et méthodes statistiques ;
- 2000s : Utilisation du World Wide Web comme corpus

3- Différents niveaux de traitement automatique du langage naturelle

Nous introduisons dans cette section les différents niveaux de traitement nécessaires pour parvenir à une Compréhension complète d'un énoncé en langage naturel. Du point de vue de l'ingénieur, ces niveaux Correspondent à des modules qu'il faudrait développer et faire coopérer dans le cadre d'une application complète de traitement de la langue naturelle. Mais il n'est pas absurde de voir également dans ces nivaux, tant ils semblent demander des connaissances et des mécanismes différents, un modèle des différents composants de la machinerie cognitive mobilisée dans la production et la compréhension du langage naturelle.

- Niveau de traitement Morpho-Lexical ;
- Niveau de traitement syntaxique ;
- Niveau de traitement sémantique ;
- Niveau de traitement pragmatique [7].

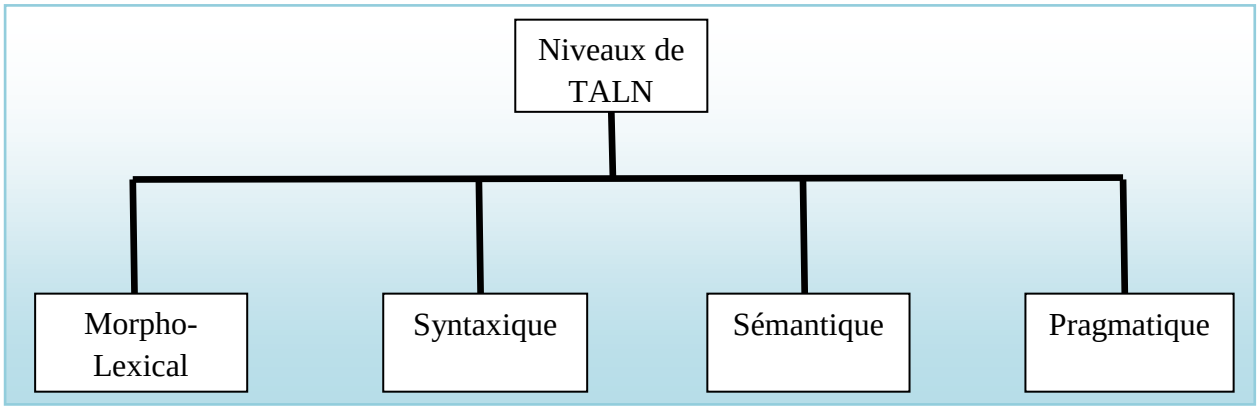


Figure1 Représentation des niveaux de traitement du langage naturel

3.1. Niveau Morphologie

La morphologie interprète comment les mots sont structurés et quels sont leurs rôles dans la phrase. Cette analyse consiste à une segmentation du texte en unités élémentaires auxquelles sont attachées des connaissances dans le système : une fois cette segmentation effectuée, ce n'est plus le texte qui est manipulé, mais une liste ordonnée d'unités. Pour le traitement d'un texte numérique : on part d'une chaîne de caractères typographiques, et on essaie de la segmenter de manière à ce que chaque partie corresponde à une unité classée dans le système [7].

Exemple : soit la chaîne de caractères « يكتب عمر الدرس»

La segmentation se fera de la manière suivante :

- U1 =يكتب
- U2 **=** عمر
- U3 =الدرس

Maintenant, on pourra associer toutes sortes d'informations aux U**i** (**i** = 1, 2, 3, ...), comme :

Exemple :

- U2=عمر
  - ✓ Informations morpho-syntaxiques : nom propre, masculin, singulier.
  - ✓ Informations sémantiques : animé, humain, prénom ...
- U1= يكتب
  - ✓ Forme lemmatisée : كتب
  - ✓ Informations morpho-syntaxiques : verbe (فعل), passé (ماضي), indicatif, 3^{ieme}personne, singulier. [7]

3.2. Niveau syntaxique

C'est une partie de la grammaire qui traite la manière dont les mots peuvent se combiner pour former des propositions et de l'enchaînement des propositions entre elles. Cela consiste à associer, à la chaîne découpée en unités, une représentation des groupements structurels entre ces unités ainsi que des relations fonctionnelles qui unissent les groupes d'unités (voir Figure2).

Reprenons l'exemple précédant : «يكتب عمر الدرس », et sa représentation morphologique:

- U1 = يكتب
- U2= عمر
- U3= الدرس [7].

Le résultat de l'analyse syntaxique pourra être par exemple l'arbre suivant :

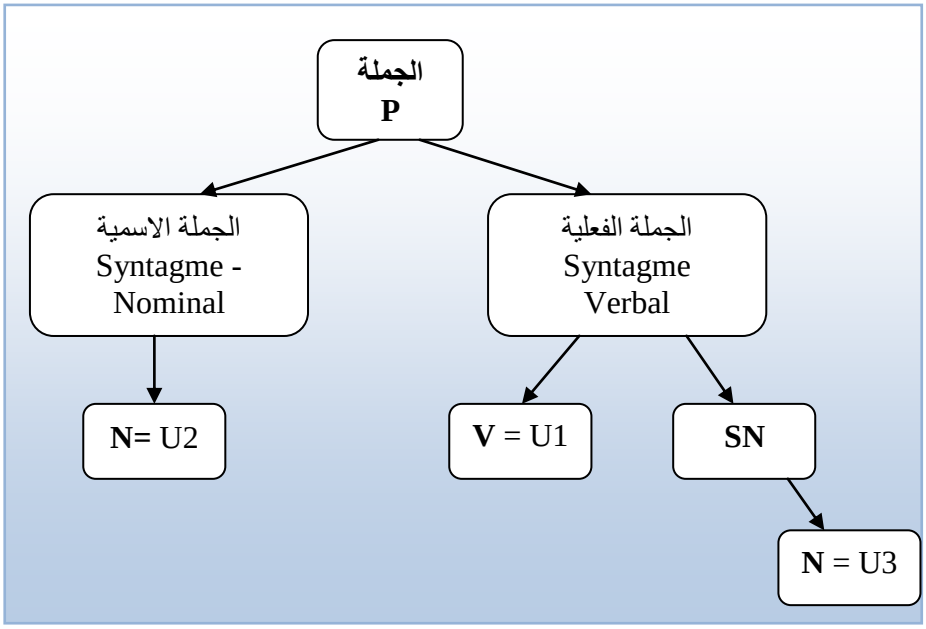


Figure2 : Arbre syntaxique de la phrase P

- **P** = «يكتب عمر الدرس»
- **SN** = عمر
- **SV** = يكتب الدرس
- **SN** = الدرس
- **N** = عمر
- **V** =يكتب
- **N** = الدرس

3.3. Niveau sémantique

L'analyse sémantique est encore beaucoup plus complexe à décrire et à formaliser que les niveaux précédemment énoncés. De ce fait, peu d'outils de traitement reste opérationnel ou du moins, concernent des applications très réduites où l'analyse sémantique se limite à un domaine parfaitement étroit ; par contre, il reste beaucoup à apprendre sur la manière de construire en grandeur réelle des analyseurs sémantiques généraux qui couvriraient la totalité de la langue arabe et seraient indépendants d'un domaine d'application particulier.

La phrase est l'unité d'analyse principale que prend en charge le traitement sémantique afin de représenter sa partie significative. Ces phrases, dont l'analyseur sémantique doit décrire le

sens, se composent d'un certain nombre de mots identifiés par l'analyse morphologique, et regroupés en structures par l'analyse syntaxique [7].

3.4. Niveau pragmatique

Ce type de traitement permet de lever les ambiguïtés qui ne peuvent pas être éliminées par le traitement sémantique, à cause de certains problèmes ayant un lien avec le contexte dans lequel la phrase est prononcé (donner un sens au mot par rapport au contexte dans lequel il se trouve), c'est-à-dire, il se charge de placer le mot dans le contexte de l'ensemble des connaissances en faisant recours à des informations hors-contexte (géographie, sport, travail, …etc.).

La séquentialité de ces traitements est une idéalisation. Dans la pratique, il est préférable de concevoir ces niveaux de traitement comme des processus coopératifs, qui échangent de l'information dans les deux sens (à la fois des niveaux «bas» vers les niveaux «hauts» ,et en sens inverse): il est ainsi souvent nécessaire de faire appel à des informations sémantiques pour trouver la «bonne» structure syntaxique d'une phrase, etc [7].

4- Domaines d'applications du TALN

Concernant les applications, la demande de TALN provient, pour dire vite, de deux tendances «lourdes»:

D'une part la nécessité de concevoir des interfaces de plus en plus ergonomiques, d'autre part la nécessité de pouvoir traiter (produire, lire, rechercher, classer, analyser, traduire) de manière de plus en plus «intelligente» les informations disponibles sous forme textuelle, de manière à pouvoir résister à leur prolifération exponentielle. Les applications de TALN (Figure 3) sont donc nombreuse set variées [3].

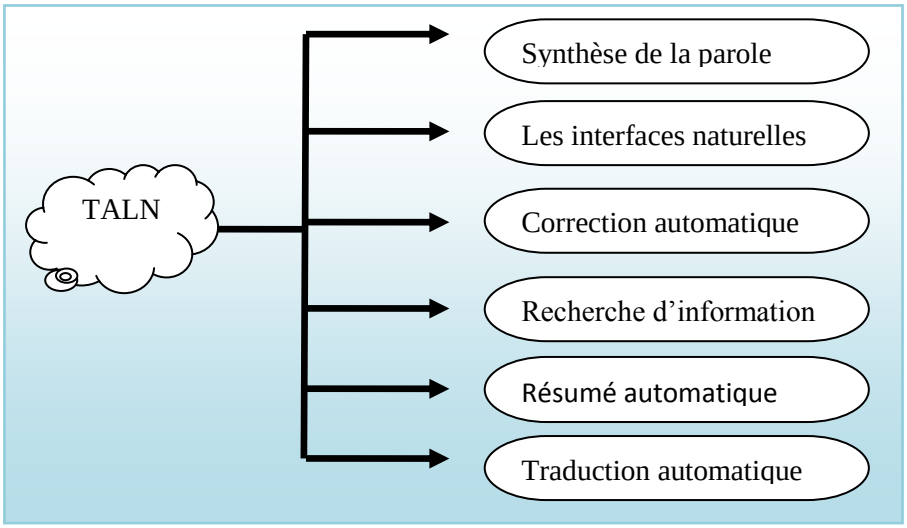


Figure3 : domaines d'applications du TALN

D'après la figure précédente en peut regrouper ces applications en trois grandes familles, qui correspondent en relation avec la production où la modification de texte, relation avec le traitement du signal, et en fin en relation avec l'extraction d'information comme suite :

A. Les applications en relation avec la production où la modification de texte

- ✓ la traduction automatique ;
- ✓ la génération automatique de textes ;
- ✓ la correction orthographique ;
- ✓ le résumé automatique de texte ;
- ✓ la reconnaissance de l'écriture manuscrite ;
- ✓ les agents conversationnels;
- ✓ la résolution d'anaphores;
- ✓ La reformulation et le paraphrase [4].

B. Les applications en relation avec le traitement du signal

- ✓ la reconnaissance automatique de la parole ;
- ✓ la synthèse de la parole ;
- ✓ le traitement de la parole [4].

C. Les applications en relation avec l'extraction d'information

- ✓ la recherche d'information et la fouille de textes ;
- ✓ la reconnaissance d'entités nommées, étant donné un texte, déterminer les noms propres, tels que des personnes ou des endroits;
- ✓ l'annotation sémantique ;

- ✓ la classification et la catégorisation de documents ;
- ✓ la détection de coréférences [4].

5. Etude de La langue arabe

L'arabe (al arabiya en transcription traditionnelle) est la langue parlée à l'origine par les Arabes. C'est une langue sémitique (comme l'akkadien et l'hébreu).Au sein de cet ensemble, elle appartient au Sous groupe du sémitique méridional .du fait de l'expansion territoriale au Moyen Âge et par la diffusion du Coran, cette langue s'est répandue dans toute l'Afrique du nord et en Asie mineure. Dire langue arabe, c'est donc parler d'un ensemble complexe dans le quel se déploient des variétés Ecrites et orales répondant à un spectre très diversités d'usages sociaux, des plus savants aux plus populaires. Mais au de là de cette diversité, les sociétés arabe sont une conscience aiguë d'appartenir à une Communauté linguistique homogène [19].

5.2Particularités de la langue arabe

Au contraire de nombreuses autres langues, l'arabe s'écrit et se lit de droite à gauche. Une autre originalité concerne l'utilisation facultative des voyelles. Les voyelles sont ajoutées au-dessus ou au-dessous des lettres, sous la forme des signes diacritiques. Elles sont utiles à la lecture et à la compréhension correcte d'un texte, car elles permettent de distinguer des mots ayant la même représentation graphique. Elles sont utiles, notamment, pour effectuer la correcte interprétation grammaticale d'un mot indépendamment de sa position dans la phrase.

5.1.1 L'alphabet arabe

L'alphabet de la langue arabe compte 28 consonnes appelées « Huruf al_Hija » Ses alphabets changent de forme en fonction de leur position dans le mot (début, milieu, fin, isolé) et se compose de deux familles contenant le même nombre de consonnes :

- Familles Solaires : contient 14 consonnes.
- Familles Lunaires : contient 14 consonnes [4].

Familles Solaires														Familles Lunaires												
ن	ل	ظ	ط	ض	ص	ش	س	ز	ر	د	ذ	ث	ت	ي	و	م	ه	آ	ق	ف	غ	ع	خ	ح	ج	ب	ا

Tableau 1: Classification des consonnes arabes

Par ailleurs, l'alphabet arabe compte 6 voyelles qui sont aussi divisées en deux groupes [4] :

- Voyelles courtes : 3 voyelles.
- Voyelles longues : 3 voyelles.

5.1.2 Catégories d'un mot arabe

Les mots arabes sont divisés en trois catégories : noms, verbes et particules (Figure 4)[4],[5].

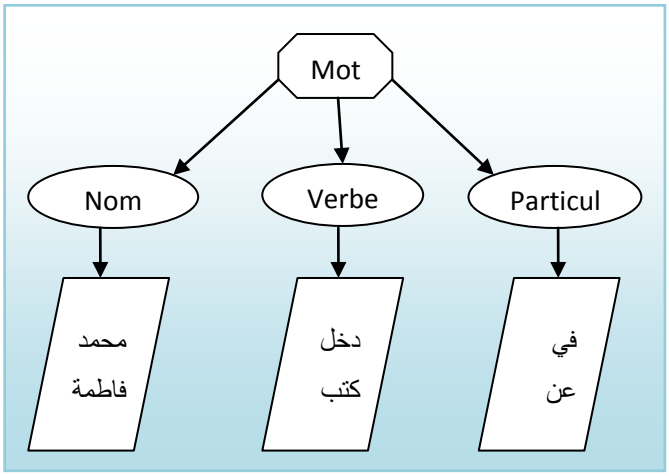


Figure4 : Catégories d'un mot arabe

❖ **Le nom**

L'élément désignant un être ou un objet qui exprime un sens indépendant du temps. En arabe les noms sont divisés en deux familles, ceux qui sont dérivés à partir d'une racine (verbale) et les autres comme les noms étrangers et certains noms fixes. La première famille est composée des tous les noms qui sont dérivés à partir d'une racine verbale. La variabilité des noms obéit à plusieurs règles, en ajoutant des morphèmes spécifiques [4] :

- Le féminin singulier : pour obtenir le nom féminin singulier, dans la majorité des cas on ajoute le lettre «ة » .

  Exemple : « مـــعلم » devient « مـــعلمة ».
- Le féminin pluriel externe : pour obtenir le nom féminin pluriel on ajoute les deux lettres « تا ».

  Exemple : « طاولـــــة » devient « طـــاولات ».
- Le masculin pluriel externe: on ajoute les deux lettres « ين » ou « ون » qui dépendent de la position du nom dans la phrase (avant ou après le verbe).

  Exemple : « مسلـــم » se transforme en « مينمسلـ » ou « مسلمـــون ».

- Le pluriel masculin, féminin et interne : c'est le cas le plus complexe en arabe, la construction de ces types des noms s'obtient en insérant des lettres au début, au milieu ou à la fin .

  Exemple : « طفـــل » se transforme en « أطفـــال » et « فصـــل » se transforme en « فصـــول »).

Comme en français, les noms en arabe assument des fonctions diverses [4]:

- ✓ Agent : celui qui fait l'action ;
- ✓ Objet : celui qui subit l'action ;
- ✓ Instrument : signifiant l'instrument de l'action ;
- ✓ Lieu : qui désigne en général un endroit ;
- ✓ Nom d'action : désigne l'action ;
- ✓ etc …

❖ **Le verbe**

Nous pouvons classer les verbes arabes selon plusieurs critères : Selon le nombre et la nature des consonnes de leurs racines, et selon leurs modèles. En classant les verbes selon le nombre des consonnes de la racine, nous aurons soit des verbes trilitères qui ont trois consonnes, soit des verbes quadrilatères, peux nombreux, qui ont quatre consonnes.

Selon le modèle et le nombre de consonnes qui constituent la structure verbale, nous avons soit des verbes nus (مُجرد) qui sont composés seulement par les consonnes de leurs racines et

des voyelles brèves, soit des verbes augmentés ou dérivés (مزيد) qui sont dérivés de trois consonnes de la racine par modification des voyelles, par redoublement de la deuxième lettre de la racine, par adjonction et même par intercalation d'affixes.

La conjugaison des verbes dépend de plusieurs facteurs :

- ✓ Le temps (accompli, inaccompli).
- ✓ Le nombre du sujet (singulier, duel, pluriel).
- ✓ Le genre du sujet (masculin, féminin).
- ✓ La personne (première, deuxième et troisième)
- ✓ Le mode (actif, passif) [6].

❖ **Les particules**

Les particules sont des lemmes invariables et en nombre limité. Ils indiquent l'articulation de la phrase. Elles sont classées selon leur champ sémantique et leur fonction dans la phrase; on en distingue plusieurs types :

- ✓ Préposition : exemple (ك ، ب ، ل ، عن، حتى)
- ✓ Particules de coordination : exemple (و، ف، ثم، أو)
- ✓ Particules interrogatives : exemple (ما، هل، أ)
- ✓ Particules d'affirmation : exemple (نعم ، بلى، أجل،)
- ✓ Particules de négation : exemple (لا، لن، لم)
- ✓ Particules distinctive : exemple (أي)
- ✓ Particules relatives : exemple (أم)
- ✓ Particules de futur : exemple (س، سوف)
- ✓ Particules conditionnelles : exemple (إن، لو)

Ces particules seront très utiles pour notre traitement, elles font partie du dictionnaire qui regroupe les mots vides.

Les particules peuvent avoir des préfixes et suffixes ce qui rajoute une complexité quant à leur identification [6].

6. Les éléments de base de la morphologie arabe

La langue arabe a une morphologie riche et différente, par rapport aux langues occidentales. L'analyse morphologique d'un mot arabe, consiste principalement à déterminer la structure générale de ce mot (Tableau 2), s'il existe, et les autres éléments utilisés pour construire ce mot (les affixes[4], les racines).

Suffixe	Racine	Préfixe
ـون	رسمـــ	يـــــــ

Tableau 2 : exemple structure d'un mot arabe

Les éléments de base de la morphologie de la langue arabe sont :

6.1 Les racines

Les racines sont à l'origine de la plupart de mots arabes. Elles sont des verbes formés de trois à cinq lettres consonne. Elles sont aux alentours de 10000 racines dont la grande

[4] Préfixes et suffuxes.

majorité (85%) sont trilatérales. Les restes sont des racines quadrilatérales ou cinq unité latérales. Une racine définit la signification fondamentale des mots dérivés en utilisant différents diacritiques et affixes avec les lettres de la racine pour créer l'inflexion de la signification [6].

6.2 Les schèmes

Le modèle arabe permet essentiellement de déterminer la structure de la plupart des mots (les noms, les verbes conjugués, etc.). Les modèles sont des déclinaisons du mot <فَعَلَ, faire>qui sont obtenus en utilisant des diacritiques ou en y ajoutant des affixes. Par exemple, le modèle <فُعِلَ, a été fait> est obtenu en utilisant les diacritiques, le modèle <مُسْتَفْعَل> est obtenu en y ajoutant le préfixe [6].

6.3 Les affixes

Les affixes sont des lettres qui s'ajoutent au début (les préfixes) ou à la fin des mots arabes (les suffixes). En général, Ils sont utilisés pour accorder aux mots des éléments syntaxiques. Ils marquent l'aspect verbal, le mode, les propriétés transitives, etc.

Les préfixes dépendent des mots auxquels ils s'attachent. En effet, la plupart des mots arabes commencent par le préfixe < ال التعريف , al altâryif, l'article de définition > qui est utilisé en tant
que terme déclaratif. Pour cela, il y a trois types des préfixes. Premièrement, les préfixes nominaux qui sont réservés pour les noms et les adjectifs. Deuxièmement, les préfixes verbaux qui sont réservés aux verbes. Et troisièmement, les préfixes généraux qui sont utilisés indépendamment de type des mots.

Il y a deux types des suffixes, les suffixes verbaux et les suffixes nominaux. Les premiers dépendent de la transitivité et de la personne conjuguée . Les suffixes nominaux indiquent la flexion casuelle du nom (nominatif, accusatif, et génitif), le genre (masculin et féminin), le nombre (singulier, duel et pluriel) [6].

6.4 Mots dérivés

Les mots dérivés sont construits à partir d'une racine en y ajoutant des affixes. La plupart des mots arabes sont considérés comme des mots dérivés, puisqu'ils sont construits à partir des racines. Ainsi, les mots qui dérivent d'une même racine ont des significations similaires. En effet, certains mots dérivés peuvent avoir la signification d'une phrase entière.

6.5 Mots isolés

Les mots isolés sont les mots qui n'ont pas des racines. Les mots sont en général, les noms propres, les noms communs et les particules.

Exemple : « انســان », « افـريقيا ».

6.6 Signes diacritique

Les signes diacritiques (Tableau 3) placés au dessus ou au dessous des lettres. On constate l'étendue du rôle que jouent les voyelles en arabe, non seulement parce qu'elles enlèvent l'ambiguïté mais aussi elles donnent l'étiquette grammaticale d'un mot indépendamment de sa position dans la phrase. Cependant, elles ne sont utilisées que pour des textes didactiques. Les textes courants rencontrés dans les journaux et les livres n′en ne comportent habituellement pas [6].

Fatha(فتحة)	Kasra(كسرة)	Dama(ضمة)	Sukun(سكون)	Chada(شدة)	Tanwin(تنوين)		
◌َ	◌ِ	◌ُ	◌ْ	◌ّ	◌ً	◌ٌ	◌ٍ

Tableau3 : signes diacritiques de l'arabe

7. Problèmes du traitement automatique de l'arabe

7.1. L'ambiguïté

En traitement automatique des langues naturelles en particule la langue arabe, le principal problème à résoudre est l'ambiguïté. Il existe différents types d'ambiguïtés. D'abord, les mots peuvent être ambigus aux niveaux lexical ou grammatical. Le mot « مغرب » est ambigu lexicalement. Il peut désigner « Maroc » en français ou encore « temps de préaire ». « كتب » quant à lui, est ambigu grammaticalement. Il peut appartenir à plusieurs catégories grammaticales différentes : verbe ou nom. Le sens de ce mot sera très différent selon sa catégorie nom = « كاتب », verbe = « كَتَبَ ». Il existe aussi des ambiguïtés qui relèvent du niveau syntaxique. Une même phrase peut avoir plusieurs sens possibles en fonction de ses interprétations syntaxiques [6], [7].

7.2. Absence des voyelles

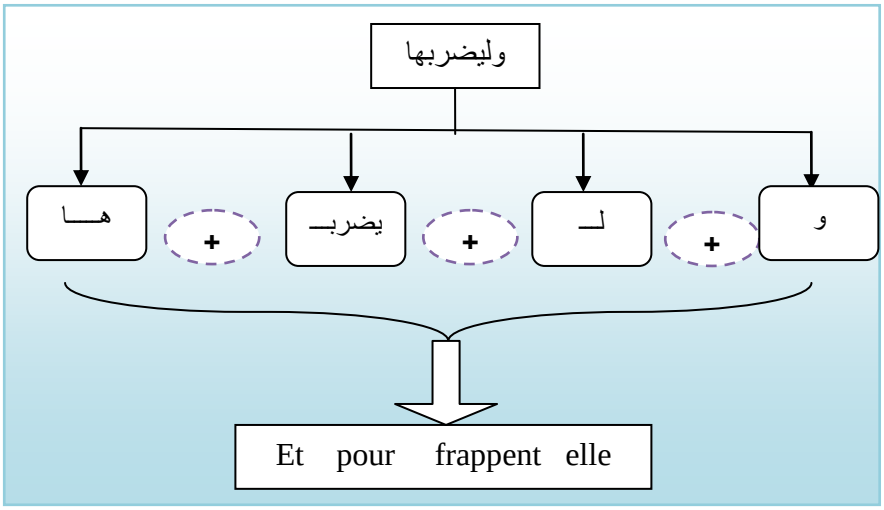
La majorité des textes en arabe sont écrits à l'aide de lettres non voyelles, Un mot sans voyelles peut générer plusieurs cas d'ambiguïtés lexicales et morphologiques. Par

exemple le mot sans voyelle « ktb » possède 17 voyellations potentielles, représentant 9 catégories grammaticales différentes [7].

7.3. Agglutination

Contrairement aux langues latines, l'arabe est une langue agglutinante ; les articles, les prépositions et les pronoms collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent, autrement dit l'ensemble des morphèmes collés les unes aux autres et constituant une unité lexicale véhiculent plusieurs informations morpho-syntaxiques ce qui engendre une ambiguïté morphologique au cours de l'analyse des mots [6], [7]**.**

Exemple : « وليضربها »



Figue5: Exemple d'agglutination « وليضربها »

8. Conclusion

Dans ce chapitre nous avons présenté l'état de l'art du TALN à partir de son histoire, niveaux de traitement, domaines d'applications, ainsi que l'étude des caractéristiques principales de la langue arabe qui sont différentes par rapport aux langues européennes.

La langue arabe se caractérise par une richesse flexionnelle, par ses signes de vocalisation et son ambiguïté, ce qui présente des nombreux défis pour des domaines divers tels que le traitement automatique du langage naturel ou aussi la recherche d'informations.

1. Introduction

A cause de la croissance rapide du nombre et du volume d'information stockées électroniquement, un grand problème se pose pour chercher et rependre d'une requête avec une manière pertinente à partir d'un ensemble de documents dans une base de documents qui s'appelle corpus. En général ce dernier n'est pas structuré et ce problème de recherche est connu sous le nom de Recherche d'Information (***RI***).

Ce chapitre a pour but de présenter le domaine de la recherche d'information. Premièrement on présent une brève histoire, puis nous présentons les concepts de base de la recherche d'information. En particulier on découvre les notions de document, de requête et de pertinence; les processus d'indexation, de recherche et de reformulation de requête; ainsi que les modèle de RI, dernièrement on parle de l'évaluation des systèmes de recherche d'information.

2. Bref historique de la recherche d'information

- Le domaine de la recherche d'information remonte au début des années 1950, peu après l'invention des ordinateurs.
- Comme plusieurs autres domaines informatiques, les pionniers de l'époque étaient enthousiastes à utiliser l'ordinateur pour automatiser la recherche des informations, qui dépassaient la capacité humaine : il y avait une explosion d'information après la deuxième guerre mondiale.
- Le nom de « recherche d'information » (information retrieval) fut donné par Calvin N. Mooers en 1948 pour la première fois quand il travaillait sur son mémoire de maîtrise [10].
- La première conférence dédiée à ce thème – International Conference on Scientific Information - s'est tenue en 1958 à Washington. On y comptait les pionniers du domaine, notamment, Cyril Cleverdon, Brian Campbell Vickery, Peter Luhn, etc [10].
- Les premiers problèmes qui intéressaient les chercheurs portaient sur l'indexation des documents afin de les retrouver. Déjà à la « International Conference on Scientific Information », Luhn avait fait une démonstration de son système d'indexation KWIC qui sélectionnait les index selon la fréquence des mots dans les documents, et filtrait des mots vides de sens en employant des « stoplistes ». C'est à cette période que le domaine

de RI est né.

- 1960-1970 : Expérimentations plus larges ont été menées. On a développé une méthodologie d'évaluation du système qui est aussi utilisée maintenant dans d'autres domaines (des corpus de test ont été conçus pour évaluer des systèmes différents).
- 1970 : Développement du système SMART. Les travaux sur ce système a été dirigés par G. Salton. Certains nouvelles techniques ont été implantées et expérimentées pour la première fois dans ce système (par exemple, le modèle vectoriel et la technique de relevance feedback). Du côté de modèle, il y a aussi beaucoup de développements sur le modèle probabiliste.
- 1980 : Les travaux sur la RI ont été influencés par l'avènement de l'intelligence artificielle. Ainsi, on tentait d'intégrer des techniques de l'IA en RI, par exemple, système expert pour la RI, etc.
- 1990 : Internet à propulser la RI en avant scène de beaucoup d'applications. La venue de l'Internet a aussi modifié la RI. La problématique est élargie. Par exemple, on traite maintenant plus souvent des documents multimédia qu'avant. Cependant, les techniques de base utilisées dans les moteurs de recherche sur le web restent identiques [10] ,[23].

3. Définitions de la recherche d'information

Toutes les définitions de la recherche d'informations et des systèmes de recherche d'informations tournent sur le même axe :

La recherche d'information est un ensemble de techniques offrant la possibilité de retrouver à partir d'une grande masse de documents ceux qui sont susceptibles correspondant aux besoins d'un utilisateur, en utilisant souvent une requête dans un langage naturel [6].

Un système de recherche d'information est un ensemble de techniques et de processus permettant à un utilisateur d'accéder à des documents qui ont une contribution dans la résolution du problème d'information qui motive sa recherche [9].

Il est constitué d'un modèle de représentation des documents du corpus[5] ainsi que la requête qui exprime le besoin de l'utilisateur, dans le but de fournir comme résultats des documents pertinents pour l'utilisateur [14].

[5] Un corpus est un ensemble de documents, artistiques ou non (textes, images, vidéos, etc.), regroupés dans une optique précise.

Dans ses définitions, il y a trois notions clés : documents, requête, pertinence.

- **Document**: Un document peut être un texte, un morceau de texte, une page Web, une image, une bande vidéo, etc. On appelle document toute unité qui peut constituer une réponse à une requête d'utilisateur.
- **Requête:** Une requête exprime le besoin d'information d'un utilisateur. Elle est en général de la forme suivante: "Trouvez les documents qui …".
- **Pertinence:** Le but de la RI est de trouver seulement les documents pertinents [16].

  la pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Cependant, la définition de cette notion complexe n'est pas simple, car elle fait intervenir plusieurs notions. basiquement elle peut être définie comme la correspondant entre un document et une requête ou encours comme une mesure d'informativité du document à la requête. Essentiellement; deux types de pertinence sont définis: la pertinence système et la pertinence utilisateur.

  - ✓ La pertinence système: est souvent présentée par un scors attribué par le système de recherche d'information (SRI) afin dévaluer l'adéquation du contenu des documents vis-à-vis de celui de la requête. ce type de pertinence est objectif et déterministe. [22]
  - ✓ Pertinence utilisateur: quant à elle, se traduit par les jugements de pertinence utilisateur sur les documents fournis par le système de recherche d'information en réponse à une requete.la pertinence utilisateur est subjective; car pour un même document retourné en réponse à une même requête, il peut être jugé différent par deux utilisateurs distincts (qui ont des centres d'intérêt différent).de plus, cette pertinence est évolutive, un document jugé non pertinent à l'instant t pour une requête peut être jugé pertinent à l'instant t+1, car la connaissance de l'utilisateur sur le sujet à évolué[22].

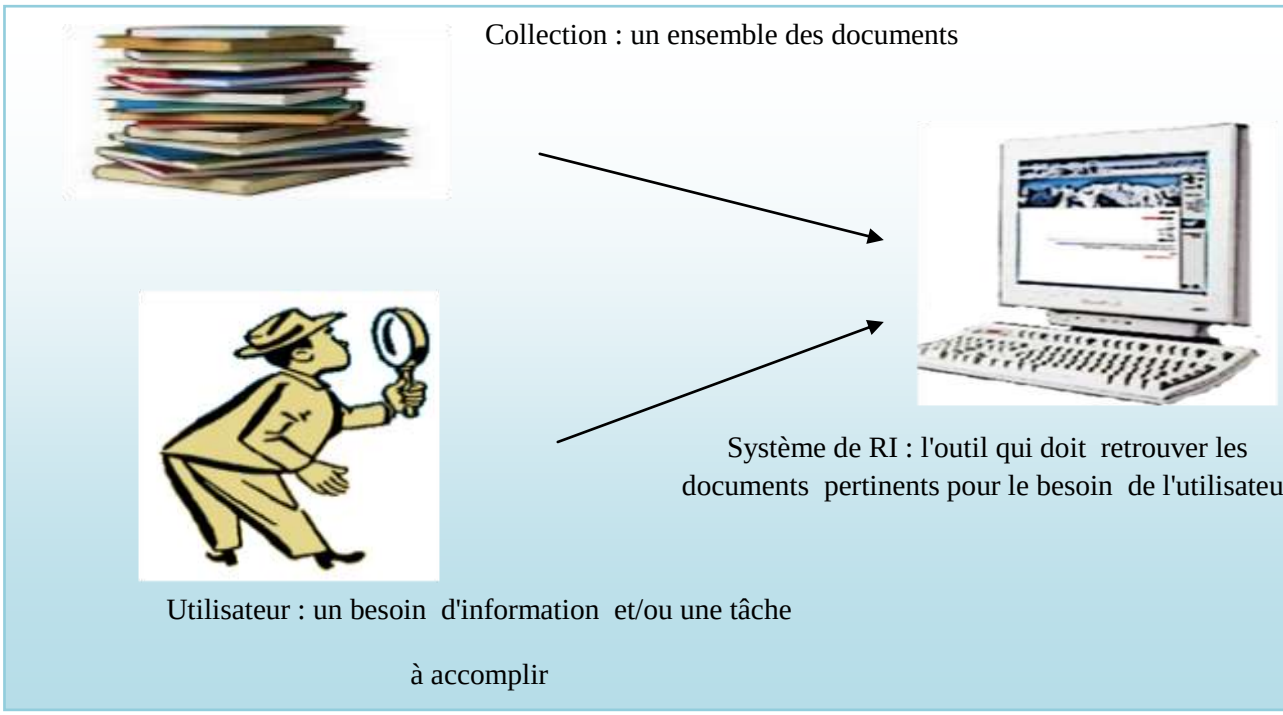


Figure 6:Les acteurs de RI

4. Processus de recherche d'information

D'après les définitions précédentes on peut déduire qu'un processus de recherche d'information permet de faire le lien entre les informations disponibles (les documents du corpus) et les besoins de l'utilisateur. Cette liaison est effectuée par un système de recherche d'information, dont le principal objectif est de correspondre au mieux entre la pertinence système et la pertinence utilisateur, tel que le document est considéré pertinent par l'utilisateur en fonction de son besoin en information, et il est considéré pertinent par le SRI sur la base de la fonction de pertinence utilisée dans ce système [15].

Les processus de RI est composé de deux étapes principales : Indexation et Recherche.

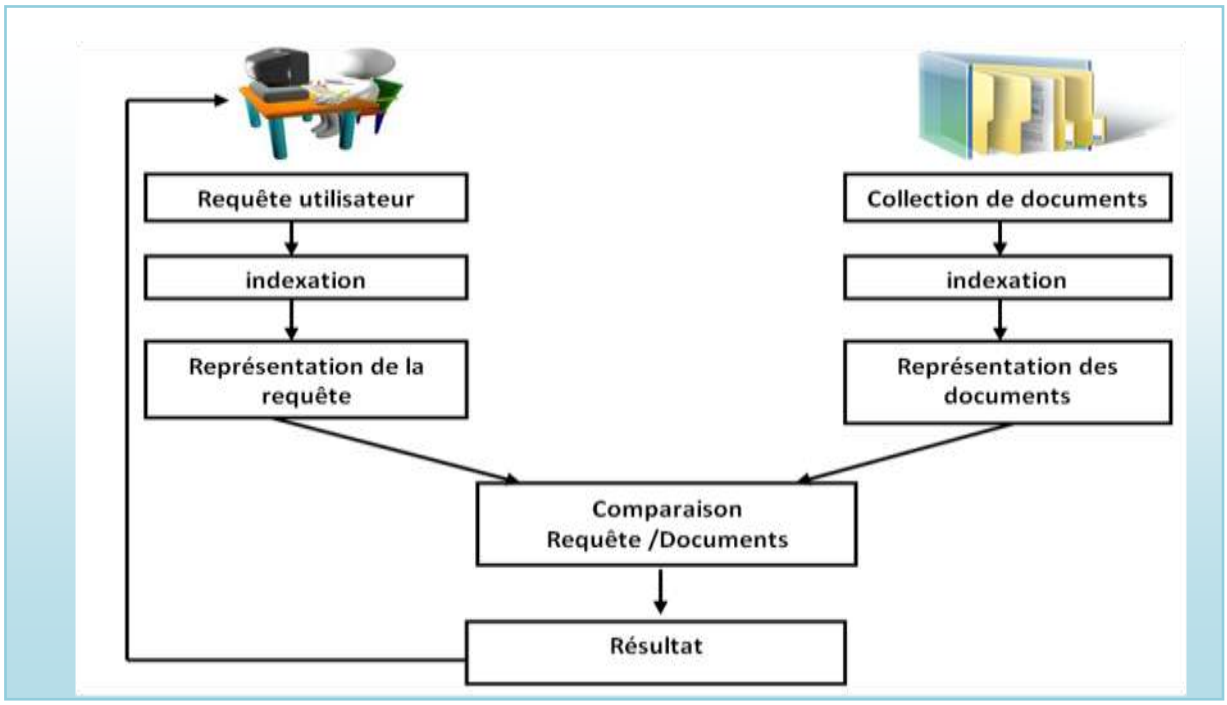


Figure7: Processus de système de recherche d'information(SRI)

L'architecture générale d'un SRI illustrée par la figure 7 fait ressortir des éléments constitutifs telque: le document, le besoin en information, la requête et la pertinence, ainsi que trois principales fonctionnalité: l'indexation, la recherche et la reformulation de la requête.

4.1. La phase d'indexation

L'indexation est une opération permettant d'extraire une représentation convenable d'un document ou d'une requête, et qui couvre au mieux son contenu sémantique. Cette représentation sémantique est généralement appelée index ou signature.

La phase d'indexation peut être effectuée de trois(03) manières:

- **Manuellement** : chaque document de la collection est analysé par un spécialiste de domaine ou un documentaliste. L'indexation manuelle assure une meilleure précision dans les documents restitués par le SRI en réponse aux requêtes des utilisateurs. Néanmoins, cette indexation un certain nombre d'inconvénients liés notamment à l'effort et le prix qu'elle exige(en temps et en nombre de personne).De plus, cette indexation est subjective, qui liée au facteur humain, différents spécialistes peuvent indexer un document avec les termes différents.il se peut même arriver qu'un spécialiste indexer différent un document, à différents moments [22], [20].
- **Semi-automatique**: la tâche d'indexation est réalisée ici conjointement par un programme informatique et un spécialiste du domaine. Le choix final des descripteurs

revient à l'indexeur humain. Dans ce type d'indexation un langage d'indexation contrôlé est généralement utilisé [22], [20].

- **Automatique**: dans ce cas, l'indexation est entièrement automatisée. Elle est réalisée par un programme informatique et elle passe par un ensemble d'étapes pour créer d'une façon automatique l'index.ces étapes sont: l'analyse lexicale, l'élimination des mots vides, la normalisation (lemmatisation ou radicalisation), la sélection des descripteurs, le calcul de statistiques sur les descripteurs et les documents(fréquence d'apparition d'un descripteur dans un document et dans la collection, la taille de chaque document,… etc.), et en fin la création de l'index et éventuellement sa compression [22], [20].

Nous détaillons ces différentes étapes ci dessous.

4.1.1. Analyse lexicale

L'analyse lexicale est le processus de conversion d'un flux de texte des documents dans un flux des mots (les mots qui peuvent être des termes d'indexations). Le principal objectif de la phase d'analyse lexicale est le traitement des chiffres, signes de ponctuation, et la casse des lettres [21].

4.1.2. Élimination des mots vides

Les mots qui sont trop fréquents dans les documents de la collection ne sont pas discriminatrice. Puisque un mot qui se produit dans 80% des documents de la collection est inutile dans la recherche. Ces mots sont souvent appelés des mots vides et sont généralement filtrés. On outre les mots vides sont des mots qui ne sont pas significatifs en langage naturel. Il existe deux méthodes pour l'élimination des mots vides :

1. Utilisation d'une liste des mots vides (stop words).

2. L'élimination des mots dépassant un certain nombre d'occurrences dans le document.

L'élimination des mots vides permet de réduire de façon considérable la taille de la structure d'indexation. Souvent cette élimination permet d'obtenir une compression de cette structure d'enivrent de 40% [21].

4.1.3. Lemmatisation

Un utilisateur spécifie souvent une requête sous forme de mots, mais seulement une variante de ces mots est présente dans un document pertinent. L'affectation d'un suffixe à un verbe est des exemples de variations syntaxiques qui empêchent une parfaite adéquation entre un mot de la requête et un mot du document. Ce problème peut être partiellement résolu par la substitution des mots avec leurs racines grammaticales.

Plusieurs méthodes de lemmatisation ont été proposées. On peut trouver par exemple la méthode qui permet l'analyse du terme et la déduction de son origine, la méthode d'utilisation d'un dictionnaire de tous les mots, et celle qui utilise un dictionnaire des terminaisons [21].

4.1.3. Pondération

La pondération est l'étape finale dans la plus part des applications d'indexation. Les termes sont pondères selon un modèle de pondération local, global ou les deux à la fois. Si les poids locaux sont utilisés, les poids des termes sont exprimés avec la mesure **TF** (Terme Frequency). Si les poids globaux sont utilisés, le poids d'un terme est donné par le facteur **IDF** (Inverse of Document Frequency) qui mesure l'importance d'un terme dans toute la collection. Cette mesure est calculée selon l'une des déclinaisons suivante :

- $log\left(\frac{N}{df}\right)$
- $log\left(\frac{N-df}{df}\right)$

Où :

- ✓ df : le nombre de documents contenant un terme donné,
- ✓ N : le nombre total de document dans le corpus.

Les systèmes de pondération les plus courants utilisent le poids local et global à la fois, tel que le poids d'un terme égal tf* idf. Cette mesure donne une bonne approximation de l'importance du terme dans les corpus des documents de tailles homogènes [21].

4.2. La phase de recherche

Après la phase d'indexation qui n'est exécutée qu'une seule fois, on passe à la phase de recherche qui se déclenche à chaque nouvelle requête. Cette étape permet l'interaction entre le système et l'utilisateur. Un utilisateur formule une requête dans un langage d'interrogation et le système affecte une traduction à cette requête dans un formalisme similaire à celui des documents du corpus lors de la phase d'indexation. Le but de cette traduction est la compréhension des besoins de l'utilisateur. Puis une fonction de calcul de la similarité est introduite pour calculer la correspondance entre la requête traduite et chaque index des documents. Les documents ayant une pertinence positive par rapport à la requête sont sélectionnés par le système d'une manière triée, tel que le premier document sélectionné

par le système est celui qui est considéré comme le plus pertinent, et le dernier document est celui qui est considéré comme le moins pertinent [9].

5. Les types des modèles de RI

Comme nous l'avons vu le but d'un SRI demeure dans sa capacité à établir une correspondance entre un document et une requête. De nombreux modèles ont été proposés en RI, ils sont généralement regroupés autour des trois familles suivantes :

- les modèles ensemblistes qui considèrent le processus de recherche comme une succession d'opérations à effectuer sur des ensembles d'unités lexicales contenues dans les documents ;
- les modèles algébriques au sein des quels la pertinence d'un document par rapport à une requête est en visage à partir de mesures de distance dans un espace vectoriel ;
- Les modèles probabilistes qui représentent la RI comme un processus incertain et imprécis où la notion de pertinence peut être vue comme une probabilité de pertinence.

La figure suivante résumé les déférents modèles de recherche d'information

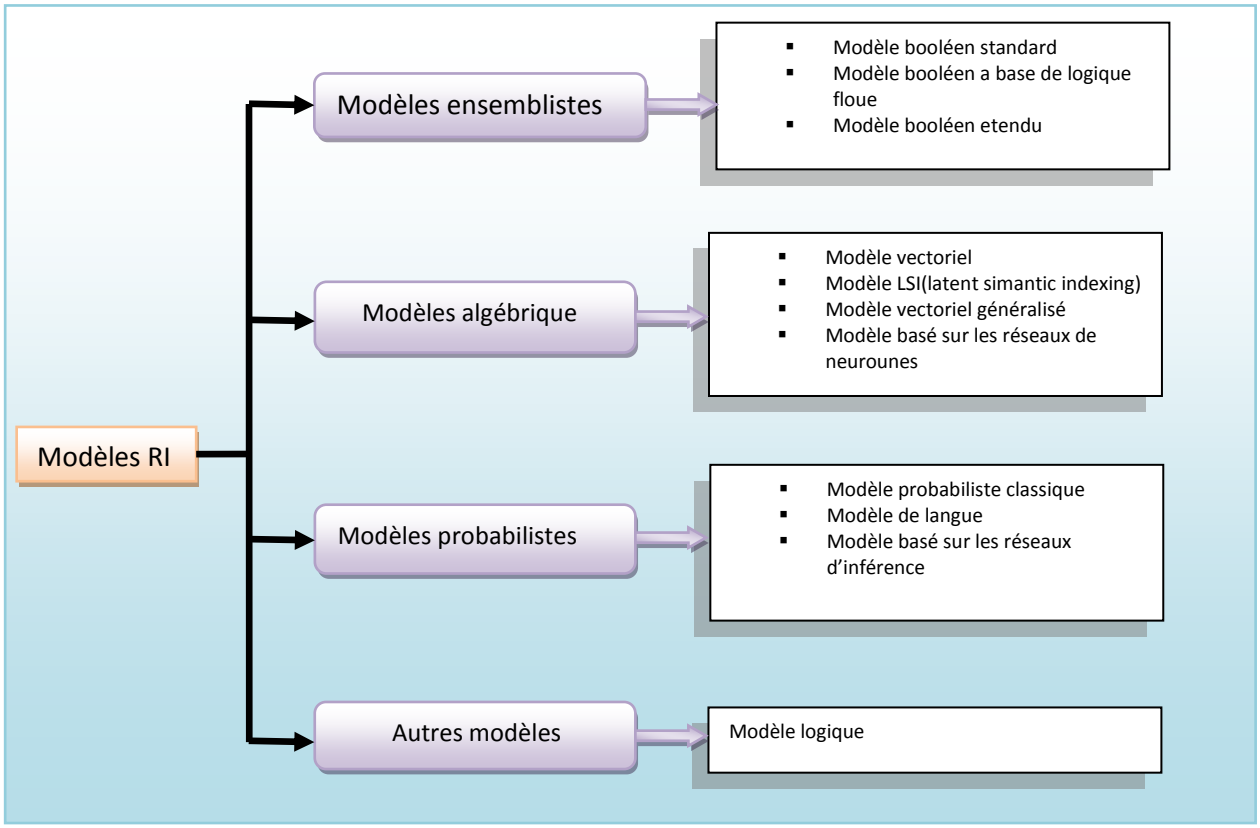


Figure 8 : Taxonomie des principaux modèles de RI

5.1. Type ensemblistes

Nous nous intéressons ici uniquement au principal représentant des modèles inspirés de la logique Booléenne et de la théorie des ensembles pour modéliser l'appariement entre une requête et les documents de la collection : le modèle booléen classique.

Le modèle booléen est le modèle le plus ancien et également le plus simple en RI .Un document est représenté par l'ensemble d'unités lexicales qu'il contient .Une requête est représentée comme une formule logique portant sur la présence ou l'absence d'unités lexicales reliées par des connecteurs (le ou « v », le et « ^ », le non « ┐ »).

Le modèle booléen fait correspondre à chaque connecteur une opération ensembliste portant les documents de la base. Sil'on note « D » la base documentaire, et « Dq » l'ensemble des documents de la base correspondant à la requête « q », on définit récursivement:

Requête	Ensemble réponse
q=t avec t un terme	Dq=Dt l'ensemble des documents contenant
$q= q_1 \wedge q_2$	$Dq = Dq_1 \cap Dq_2$
$q= q_1 \vee q_2$	$Dq = Dq_1 \cup Dq_2$
$q= q_1 \neg q_2$	Dq_1 n Dq_2

Tableau 4 : les opérations logique en modèle booléen

Ainsi , une requête : « orange ^ (ville v cité) ^ (┐ (réseau v opérateur v mobile)) »retourne à l'utilisateur le documents contenant obligatoirement l'unité lexicale "orange" et l'un des deux unités lexicales "ville" ou "cité" mais qui ne contiennent en aucun cas les unités lexicales "réseau"," opérateur" et" mobile" .Les limites de ce modèle résultent directement de la représentation choisie. Ainsi, étant donné à un document ;

La requête est soit vraie soit fausse. En termes ensemblistes, cela se traduit par la discrimination d'un document à partir de l'absence ou la présence d'une seule unité lexicale dans ce dernier [19].

5.2. Type algébriques

Couramment employé en RI, les modèles algébriques considèrent les documents et les requêtes comme faisant partie d'un même espace vectoriel, et leur appariement est fait suivant une mesure algébrique de similarité. Parmi les différentes variantes de ce type de modèle, le plus connu est le modèle vectoriel [22].

5.3. Type probabilistes

Les modèles probabilistes, dont une présentation très complète, tentent quant à eux

De modéliser la notion de pertinence. Le modèle probabiliste représente la probabilité de la pertinence d'un document « D » par rapport à une requête « R ». Le but de cette fonction de similarité dans ce modèle est d'essayer de séparer les documents pertinents des non pertinents au sein d'une collection. L'idée de base, dans ce modèle, est de tenter de déterminer les probabilités P(R∩D) et P(NR U D) pour une requête donnée.

Cette probabilité signifie : si on retrouve le document D, quelle est la probabilité qu'on obtienne l'information pertinente et non pertinente. Énonce que la présentation des documents à l'utilisateur dans l'ordre décroissant des probabilités est optimale dans un cadre de RI. Selon ce principe, les documents retournés sont ceux dont la probabilité de pertinence est supérieure à la probabilité de non pertinence [22].

6. Les modèles de recherche d'information

Un modèle de RI fournit une interprétation théorique de la notion de pertinence. Plusieurs modèles de RI on été proposés dans la littérature, ils s'appuient sur des cadres théorique différents; théorie des ensembles, algèbre, probabilités, …etc. globalement on distingue trois principales catégories de modèles: modèles booléens, modèles vectoriels, et modèles probabiliste

6.1. Modèle booléen

Les premiers SRI développés sont basés sur le modèle booléen, même aujourd'hui beaucoup de systèmes commerciaux (moteurs de recherche) utilisent booléen. Cela est du à la simplicité et à la rapidité de sa mise en œuvre.

Le modèle booléen est basé sur la théorie des ensembles et l'algèbre de boole. dans ce modèle, un document d est représenté par un ensemble de mots clés (termes) ou encors un vecteur booléen.la requête q de l'utilisateur est représenté par une expression logique, composée de termes reliés par des opérateurs logique: ET(∨),OU(∨),et NON(¬).

L'appariement entre une requête et un document est un appariement exact, autrement dit si un document implique au sens logique la requête alors le document est pertinent .sinon il est considéré no pertinent. La correspondance entre document et requête est déterminée comme suit [22]:

$$R(d, q_1) = \begin{cases} 1 \; si \; q_1 \in d : q_1 est \; un \; terme \; de \; q \\ 0 \; sinon \end{cases} \quad (2,1)$$

$$R(d, q_1 \wedge q_2) = \begin{cases} 1 \; si \; R(d, q_1) = 1 \; et \; R(d, q_2) = 1 \\ 0 \; sinon \end{cases} \quad (2,2)$$

$$R(d, q_1 \vee q_2) = \begin{cases} 1 \; si \; R(d, q_1) = 1 \; et \; R(d, q_2) = 1 \\ 0 \; sinon \end{cases} \quad (2,3)$$

$$R(d, \neg q_1) = \begin{cases} 1 \; si \; R(d, q_1) = 0 \\ 0 \; sinon \end{cases} \quad (2,4)$$

6.1.1. Avantages

- Le modèle est transparent et simple à comprendre pour l'utilisateur :
  - ✓ Pas de paramètres "cachés" ;
  - ✓ Raison de sélection d'un document claire : il répond à une formule logique ;
- Adapté pour les spécialistes (vocabulaire contraint).

6.1.2. Inconvénients

- Il est difficile d'exprimer des requêtes longues sous forme booléenne ;
- Le critère binaire peu efficace ;
  - ✓ Il est admis que la pondération des termes améliore les résultats ;
- Il est impossible d'ordonner les résultats ;
  - ✓ Tous les documents retournés sont sur le même plan ;
  - ✓ L'utilisateur préfère un classement lorsque la liste est grande. [18]

6.2. Modèle vectoriel

C'est un autre modèle souvent utilisé. Il représente les documents et les requêtes comme

Vecteurs de poids dans un espace multidimensionnel, dont les dimensions sont les termes utilisés

Pour construire un index qui représente les documents [18].

La création d'un index implique une lecture lexicologique pour identifier les termes significatifs, où l'analyse morphologique ramène les différentes formes de mot aux «lemme»[(6)] communs, et l'occurrence de ces lemmes est calculée. Des substituts de requête et de document sont comparés selon leurs vecteurs. Par exemple, Soit l'espace vectoriel suivant:

<t1, t2, t3, …, tn>

Un document et une requête qui peut représentés comme suit:

[(6)] lemme :c'est la forme canonique d'un mot.

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

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

Ainsi, a_i et b_i correspondent aux poids du terme t_i dans le document et dans la requête. [18]

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

Mesures	Formules
Le produit scalaire	$RSV(q,\ d_i\ ) = \sum_{j=1}^{\lvert T\rvert} w_{qj} \,.\, w_{ij}$
La mesure de cosinus	$RSV(q, d_i) = \frac{q.d_i}{\lvert\lvert q\rvert\rvert.\lvert\lvert d_i\rvert\rvert} = \frac{\sum_{j=1}^{\lvert T\rvert} w_{qj} \,.\, w_{ij}}{\sqrt{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2}\ \sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2}$
La mesure de Dice	$RSV(q, d_i) = \frac{2 \times \sum_{j=1}^{\lvert T\rvert} w_{qj} \,.\, w_{ij}}{\sqrt{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2} + \sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2}$
La mesure de Jaccard	$RSV(q, d_i) = \frac{\sum_{j=1}^{\lvert T\rvert} w_{qj} \,.\, w_{ij}}{\sqrt{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2} + \sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2 - \sum_{j=1}^{\lvert T\rvert} w_{qj} \,.\, w_{ij}}$

Tableau 5: les mesures de similarité utilisé dans le modèle

Sauf la première formule, toutes les autres sont normalisées [7].

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

6.2.1. Avantages

Le modèle vectoriel a les avantages suivants :

- Le langage de requête est plus simple (liste de mot clés) ;
- Les performances sont meilleures grâce à la pondération des termes ;
- Le renvoi de documents à pertinence partielle est possible ;
- La fonction d'appariement permet de trier les documents [18].

[7] c'est-à-dire qu'elles donnent une valeur dans [0, 1].

6.2.2. Inconvénients

Cependant, ce modèle a les inconvénients suivants :

- ❖ Le modèle considère que tous les termes sont indépendants ;
- ❖ Le langage de requête est moins expressif ;
- ❖ De temps en temps l'utilisateur ne sait pas pourquoi un document est retourné par le Système [18].

6.3. Modèle probabiliste

Le modèle probabiliste aborde le problème de la recherche d'information dans un cadre probabiliste. Il a été proposé au début des années 1960. Il est basé sur le principe de rang de probabilité, qui déclare qu'un système de recherche d'information est censé classer les documents basés sur leur probabilité de pertinence à la requête. Le principe tient compte qu'il y a une incertitude dans la représentation des besoins d'informations et des documents. Estimation de la probabilité de pertinence d'un document par rapport à une requête.

R : D est pertinent pour Q

¬R : D n'est pas pertinent pour Q(variables indépendantes ,deux ensembles de documents séparés)

Le but : estimer

– P(R/D) : probabilité que le document D soit contienne de l'information pertinente pour Q

– P(¬R/D)

$$si\ \frac{p(\frac{R}{D})}{p(\frac{\neg R}{D})} > 1\ ou\ si\ log\frac{p(\frac{R}{D})}{p(\frac{\neg R}{D})} > 0\ alors\ D\ est\ pertinent \qquad (2,5)$$

Rappel du théorème de **Bayes** :

$$p\left(\frac{A}{B}\right) = \frac{p\left(\frac{B}{A}\right).p(A)}{p(B)} \qquad (2,6)$$

• On ne sait pas calculer *P(R/D)*, mais on peut calculer *P(D /R)*

$$p\left(\frac{R}{D}\right) = \frac{p\left(\frac{D}{R}\right).p(R)}{p(D)} \qquad (2,7)$$

- $p\frac{D}{R}$: Probabilité d'obtenir D en connaissant les pertinents
- $p(R)$: probabilité d'obtenir un document pertinent en piochant au hasard
- $p(D)$: probabilité de piocher D au hasard

• En utilisant l'hypothèse d'indépendance des termes :

$$p\left(\frac{D}{R}\right) = \prod_{i=1}^{n} p(t_i \in \frac{D}{R}) \qquad (2,8)$$

• Pour estimer les probabilités sur les termes, on peut utiliser des requêtes déjà résolues (apprentissage) puis des pondérations [17].

6.3.1. Avantages

Les approches probabilistes ont les avantages suivants :

- Elles fournissent aux utilisateurs un rang de pertinence des documents recherchés. Par conséquent, elles leur permettent de contrôler le rendement en plaçant un seuil de pertinence ou en spécifiant un certain nombre de documents à afficher ;
- il peut être plus facile de formuler les requêtes parce que les utilisateurs ne doivent pas apprendre un langage d'interrogation et peuvent utiliser la langue naturelle [18].

6.3.2. Inconvénients

Cependant, les approches probabilistes ont les inconvénients suivants :

- Ils ont une puissance expressive limitée. Par exemple, l'opération NON ne peut pas être représentée parce que seulement des poids positifs sont utilisés ;
- Le modèle probabiliste est limité par l'absence de la structure qui exprime les caractéristiques linguistiques importantes telles que les expressions. Il est également difficile d'exprimer les contraintes de proximité, or cette caractéristique est d'une grande utilité pour les chercheurs expérimentés ;
- le calcul des scores de pertinence peut être coûteux [18].

6.4.Modèles de langue

Les modèles statistiques de langue sont exploités avec beaucoup de succès dans divers domaine: la reconnaissance de la parole, la traduction automatique, la recherche d'information …etc.

-ne pas tenter de modéliser directement la pertinence

-estimer la probabilité P(Q|D) la probabilité d'avoir la requête sachant le document, i.e. estimer la probabilité que la requête soit générée à partir du document

-repose sur l'idée que l'utilisateur, lorsqu'il formule sa requête, a une idée du document idéal qu'il souhaite retrouver et que sa requête est formulée pour retrouver ce document idéal

-idée formulée dès les années 60 dans les premiers travaux sur la RI probabiliste (Maron)

-séquence s=m1 ... mn .

$$p(s) = p(m_1 \dots m_n) = \prod_{i=1}^{n} p\left(\frac{m_i}{m_1 \dots m_{i-1}}\right) \qquad (2,9)$$

-approximation: dépendance limitée à k mots

-un contexte de (k-1) mots précédent est suffisant pour estimer la probabilité d'un mot

$$p(s) = p(m_1 \dots m_n) = \prod_{i=1}^{n} p(\frac{m_i}{m_{i-k+1} \dots m_{i-1}}) \qquad (2,10)$$

-modèles n-grams[8] : les plus utilisés sont pour n=1,2,3

6.5. Modèle LSI (Latent Semantic Indexing)

Le modèles Latent Semantic Indexing (LSI) consiste à associer et établir des relations entre les mots clés d'un corpus afin que les systèmes de recherche d'information puissent plus facilement identifier la thématique des documents du corpus.

L'approche consiste à profit de la structure évoluée implicite dans la relation entre les termes et les documents « structure sémantique » afin d'améliorer la détection des documents pertinents sur la base des termes trouvés dans les requêtes.

– Propose d'étudier les "concepts" plutôt que les termes, car ce sont eux qui relaient les idées d'un texte.

– Lie les documents entre eux et avec la requête

– Permet de renvoyer des documents ne contenant aucun mot de la requête

– Moins de dimensions [18] .

7. Critères d'évaluation des SRI

Dans les systèmes de recherche d'information, les documents retournés ne sont pas toujours absolument pertinents, donc ces systèmes ont besoin d'une évaluation qui nous permet de déterminer les performances du modèle, d'estimer ces caractéristiques, et de fournir les éléments de comparaison entre modèles [15]. Une telle évaluation est généralement basée sur un jeu de test, et une mesure d'évaluation. Dans le jeu de test on utilise une collection de documents, un ensemble d'exemple de requêtes, et un ensemble de documents pertinent associé à chaque requête, spécifié par un spécialiste. Ce dernier ensemble nous permet de comparer les documents vraiment pertinents à notre requête, et les documents pertinents retournés par le système.

On a deux mesures communément utilisées pour évaluer un système de recherche d'information sont le taux de précision et celui de rappel.

7.1. Rappel

Le pourcentage de tous les documents pertinents trouvés par le système de recherche.

[8] n-gramme : une sous-séquence de n éléments extraite d'une séquence donnée.

7.2. Précision

Le pourcentage de documents retournés par le système qui sont pertinents [17].

Ces deux mesures peuvent être définies par:

$$\text{précision} = \frac{\text{nb de documents pertinents retrouvés}}{\text{nb de documents retrouvés}} \qquad (2,11)$$

$$\text{rappel} = \frac{\text{nb de documents pertinents retrouvés}}{\text{nb dedocuments pertinents}} \qquad (2,12)$$

Deux mesures complémentaires au rappel et précision ont été définies, il s'agit du bruit et du silence.

- Le bruit: la mesure d'évaluation bruit est une notion complémentaire à la précision, elle est définie par B=1-P ou p est la précision du SRI.
- Le silence: la mesure d'évaluation bruit est une notion complémentaire à le rappel, elle est définie par S=1-R ou R est le rappel du SRI.

Courbe de Rappel-Précision: un système idéal devrait retourner tous les documents pertinents et que les documents pertinents; c'est-à-dire un taux de précision et de rappel égale 100%. Cette situation ne se pas dans un système réel car le taux de précision et de rappel sont antagonistes. En effet, lorsque la précision augmente, le rappel diminue et inversement. Ainsi pour mesurer les performances d'un système il faut utiliser les deux mesures conjointement. Cela est réalisé en calculant la paire des mesures (taux de rappel, taux de précision) à chaque document restitué [22].

Exemple :

Nous considérons par exemple un requête pour existe cinq (5) documents pertinents dans le corpus. Le tableau illustre le calcul de la précision et de rappel pour les dix(10) premiers documents retournés par un SRI. La lettre (P) précision que le document est pertinent.

Rang du document renvoyé	pertinence	rappel	Précision
Document 1	P	0,17	1
Document 2		0,17	0,5
Document 3	P	0,33	0,66
Document 4		0,33	0,5
Document 5	P	0,5	0,6
Document 6		0,5	0,5
Document 7	P	0,67	0,75
Document 8		0,67	0,5
Document 9		0,67	0,44
Document 10	P	0,83	0,5

Tableau 6: exemple de calcul de Rappel et Précision pour une requête

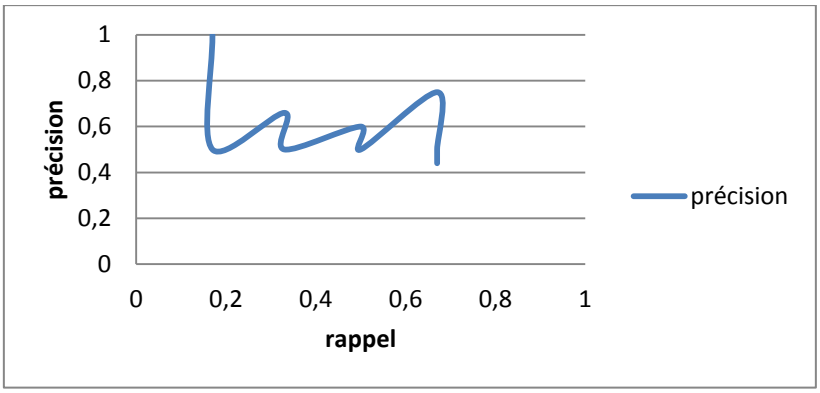


Figure 9: courbe de rappel et précision

8. Conclusion

Le but de ce chapitre était de présenter le domaine de la recherche d'information, de décrire plus Particulièrement les principales étapes à savoir l'indexation et la recherche, et d'introduire les principaux modèles sur les quels se basent les SRI, et de présenter les méthodes d'évaluation à optées pour attester de la validité des mécanismes implémentés au cœur de ces systèmes.

1. Introduction

Après avoir présenté dans le chapitre précédent le concept de recherche d'information ainsi que les différents modèles utilisés, on passe maintenant à la phase de conception générale de notre travail.

L'objectif de ce chapitre est de décrire les déférentes composantes nécessaires pour la réalisation de l'architecture générale de notre système IRYSA. En suite, on va donnant la description et le déroulement de chaque composant et leur fonctionnement.

2. L'architecture générale du système IRYSA

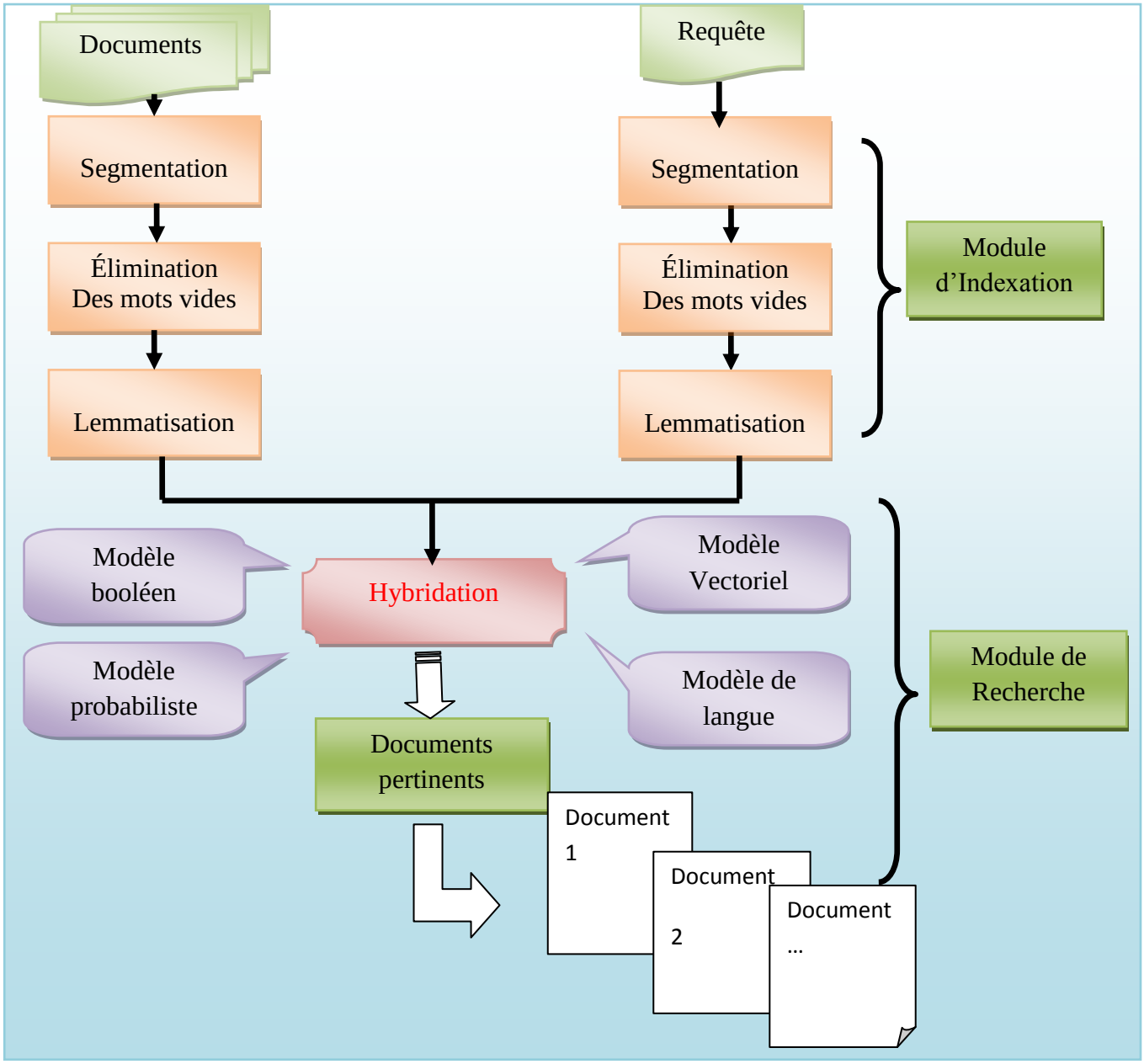


Figure 10 *:* l'architecture générale de système IRYSA

La figure (10) précédent présente l'architecture générale de notre système de recherche d'information « IRYSA »(IRYSA: Intégration d'un lemmatiseuRe arabe dans le cadre de sYSteme de recherche d'informAtion), on passe maintenant a la section suivante pour explique avec plus de précision le rôle de chaque composants.

3. Description de l'architecture

L'architecture de notre système est contient deux éléments principaux sont : l'entré qui est la requête et la sortie qu'est la collection d'information (corpus).

Ainsi que notre système est décomposé en deux modules principaux et complémentaires qui sont :

- ✓ module d'indexation
- ✓ module de recherche

a) **Le module d'indexation** : permet la représentation des termes de la collection documentaire dans un format où l'information textuelle est prise en considération.

b) **Le module de recherche** : exploite la même structure utilisée dans l'indexation des documents du corpus pour indexer la requête, afin d'avoir une structure commune entre l'ensemble des documents du corpus et les besoins de l'utilisateur. dans ce module on a essayé de proposer une formule pour le calcul de la similarité, en exploitant le contenu textuel et la relation de voisinage entre les termes des documents et celui de la requête.

3.1 Le module d'indexation d'IRYSA

L'entrée essentielle est les documents textuelles, nous avons extraire les termes significatifs des documents.

Le traitement consiste à choisir les termes significatifs ou les représentants des documents avec un ensemble d'information additionnelle à savoir:

- La fréquence d'apparition du terme dans le document;
- Identifiant du document ou le terme existe.

3.1.1 Analyse lexicale

Dans cette phase on va vérifier que les termes appartiennent au lexique arabe.

Exemple :

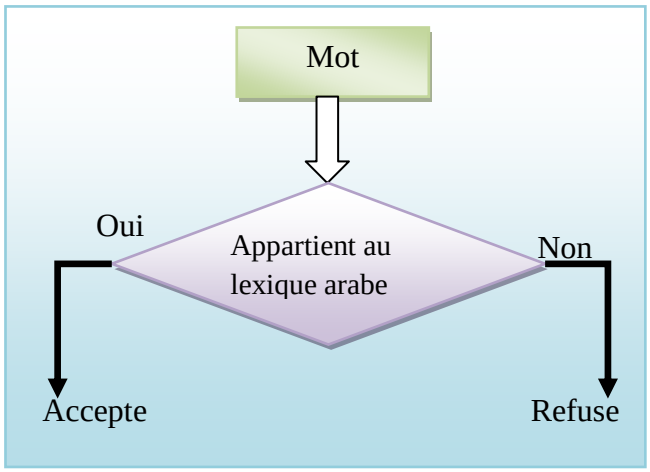


Figure 11 : Analyse lexicale propose par IRYSA

3.1.2. Lemmatisation

Cette phase consiste à représenté les différentes variantes du terme par un format unique appelé lemme ou racine. Ce qui pour effet de réduire la taille de l'index. Plusieurs stratégies de lemmatisation sont utilisées: la table de correspondance, l'élimination de l'affixe.

Dans notre système nous utilisons le lemmatiseur de texte arabe qui s'appel « lemmatiseur de khoja ».

Cet exemple exprime comment le « Stemmer Khoja » extraire le lemme d'un mot surtout les mots composés (suffixe, préfixe).

Exemple :

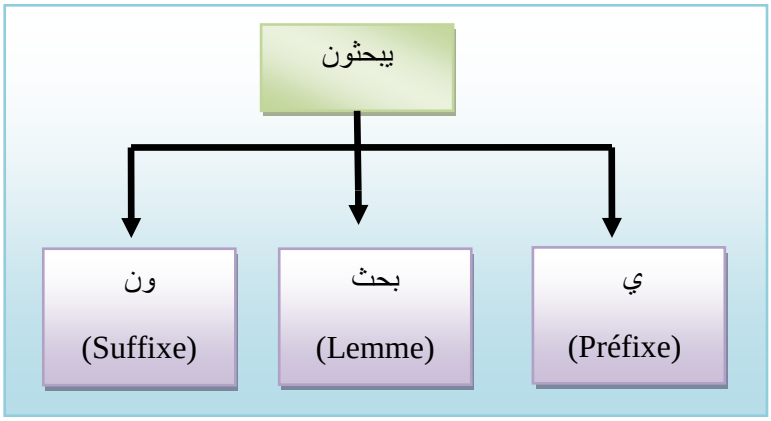


Figure12 : phase de lemmatisation

Lemmatiseur de khoja :

Khoja a proposé une méthode qui consiste à enlever les affixes après une première étape de normalisation. Ensuite, le résultat est comparé avec une liste de modèles. Si une correspondance est trouvée, les lettres représentant la racine dans le modèle sont extraits. Ensuite, la racine ainsi extraite est validée dans un dictionnaire [24], la figure suivant présente l'organigramme de lemmatiseur khoja.

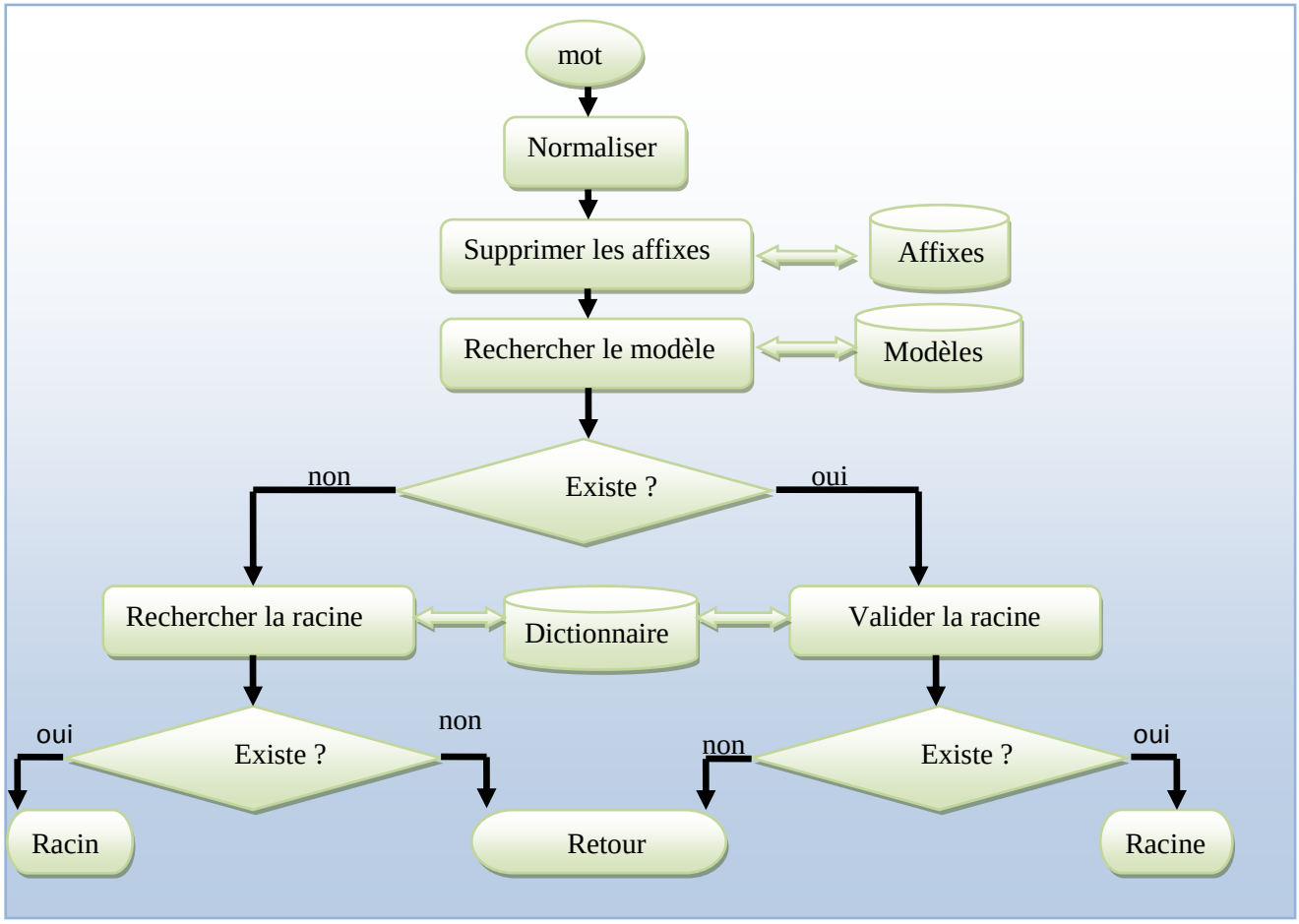


Figure 13 : organigramme lemmatiseur khoja

3.1.3. Pondération des termes

Dans l'étape de pondération on affecte à chaque terme d'index extrait lors du traitement des documents un poids qui permet de déterminer l'importance du terme au niveau du document dans lequel il est apparu et sa discrimination dans les documents de corpus.

Au niveau de notre système on a utilisé la formule « **tf.Idf** » dans le calcul des poids. Cette formule est généralement utilisée pour le calcul des poids des termes dans les documents.

La formule « ***tf*** » est définie comme suit :

$$tf_{di} = \frac{freq_{di}}{N_d} \qquad (3,1)$$

Tel que :

- $freq_{di}$: désigne la fréquence du terme t_i dans le document ***d.***
- N_d : désigne le nombre total des termes de le document ***d.***

La formule « ***idf »*** est définie comme suit :

$$idf = \frac{n}{N_d} \qquad (3,2)$$

3.1.4. Génération de l'index

Dans cette dernière étape on à essaye de construire la représentation finale de l'index associé à l'ensemble des documents de corpus. Cette représentation offre une organisation simple à l'ensemble des termes de l'index, pour qu'on puisse affecter une recherche facile et rapide au niveau de la phase de recherche.

L'index dans notre module d'indexation prendra la forme d'un fichier inverse associant à chaque terme de l'index l'ensemble des documents dans lequel il est apparu.

3.2 Le module de recherche

Un modèle de recherche d'information fournit une interprétation théorique de la notion de pertinence. Plusieurs modèles de RI on été proposés dans la littérature, ils s'appuient sur des cadres théoriques différent.

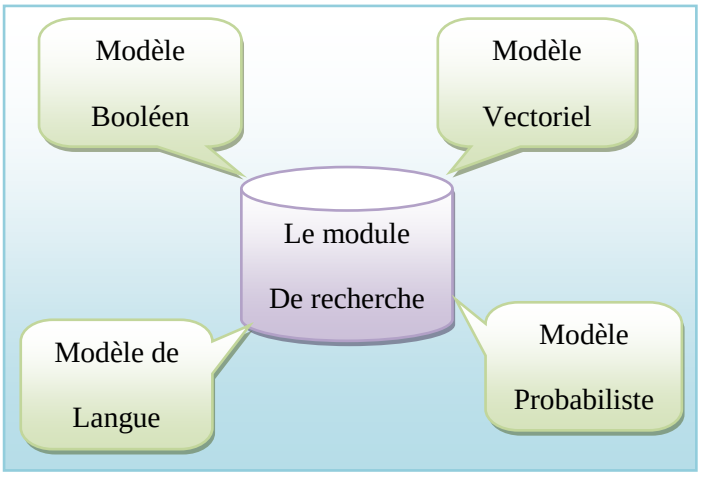


Figure 14 : présentation du module de recherche

L'algorithme de la phase de recherche suite les étapes suivantes :

1. Reformulation de la requête;

2. Normalisation des termes de la requête ;
3. Normalisation des termes des documents ;

Cas 1 : Quand la langueur de la requête est un (1 ; un seul mot) alors il suffît de marquer de présence de se terme comme suit :

- ✓ Si le terme de requête est présent dans le document 1
- ✓ Si le terme de requête est absent dans le document 0

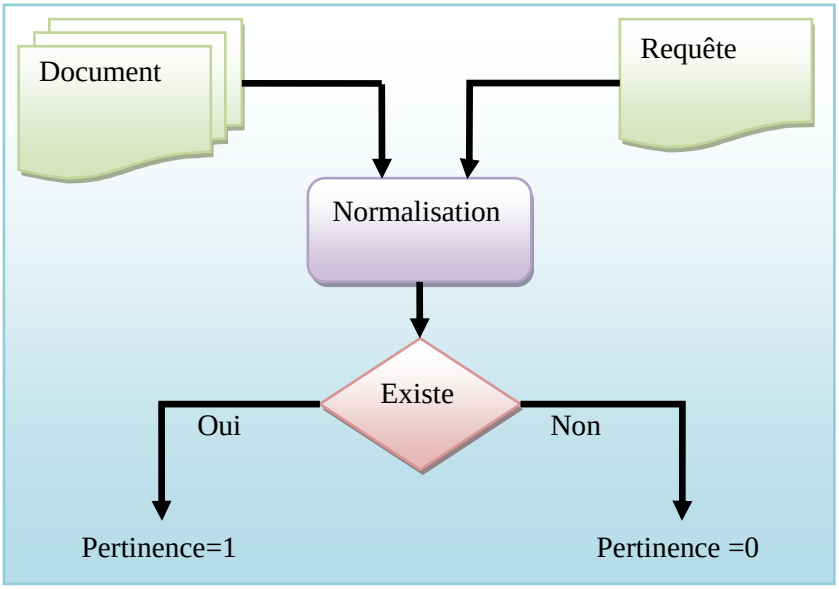


Figure 15: la présence d'un mot dans un document

Puis en va appeler la fonction de pondération des termes précédent « **tf .idf** », en suit en va calculer la similarité entre la requête est chaque document de la collection des documents. Dans ce cas en garantie l'existence de la requête dans le document puis en on calculs la similarité d'après la fréquence des termes.

Cas 2 : Si la requête contient deux mots ; alors en va suive l'algorithme suivant :

1. Reformulation de la requête;
2. Normalisation des termes de la requête ;
3. Normalisation des termes des documents ;
4. Calculé la fréquence de chaque mot ;

Dans ce cas en utilisant la notion de n-gramme

Exemple :

Considérons la séquence « s » composée des mots suivants : m_1, m_2,…, m_n. La probabilité « P(s) » peut être calculée comme suit :

$$P(s)=\prod_{i=1}^{l} p(m;\backslash m_{1\ldots} m_{i-1}) \qquad (3,3)$$

En particulier, les modèles souvent utilisés sont les modèles 'uni-gramme', 'bi-gramme' et 'tri-gramme' comme suit :

- Uni-gramme :$P(s)=\prod_{i=1}^{l} P(m_i)$ (3,4)
- Bi-gramme :$P(s) = \prod_{i=1}^{l} P(m_i \mid m_{i-1})= \prod_{i=1}^{l} \frac{P(m_{i-1}\, m_i\ )}{P(m_{i-1})}$ (3,5)
- Tri-gramme :$P(s) = \prod_{i=1}^{l} P(m_i \mid m_{i-2} m_{i-1})= \prod_{i=1}^{l} \frac{P(m_{i-2}\, m_{i-1}\, m_i\ )}{P(m_{i-2}\, m_{i-1})}$ (3,6)

Ce que l'on doit estimer, sont les probabilités P(mi) (un-gramme), P(mi-1m) (bi-gramme) et P(mi-2mi-1 mi) (tri-gramme) pour la langue. Cependant, il est difficile d'estimer ces probabilités pour une langue dans l'absolue. L'estimation ne peut se faire que par rapport à un corpus de texte « C ». Si le corpus est suffisamment grand, on peut faire l'hypothèse qu'il reflète la langue en général.

On suppose un petit corpus contenant 10 mots, avec les fréquences comme montrées dans la Table.

Mot	ذهبت	الطالبة	الصغيرة	الى	المدرسة	درست	الدروس	جميعها	نجحت	طالبتنا
Fréq	3	2	2	4	2	1	1	2	1	2
P(•|C)	0,15	0,1	0,05	0,2	0,1	0,05	0,05	0,1	0,05	0,1

Tableau 7: exemple les fréquences des termes d'un document

En utilisant l'estimation de vraisemblance maximale, nous obtenons les probabilités comme illustrées dans la table (la fréquence totale de mots dans ce corpus est |C| = 20).

En utilisant ces probabilités estimées, nous pouvons calculer la probabilité de construire la séquence s = « ذهبت الطالبة الصغيرة المدرسة » dans cette langue comme suit :

$$P(s)\approx P(s|C)=P(\text{ذهبت}|C)*P(\text{الطالبة}|C)*P(\text{الصغيرة}|C)*P(le|C)*P(\text{المدرسة}|C)=$$
$$0,15*0,1*0,05*0,15*0,1$$

Dans ce cas comme nous avons voire que pour une terme de poids nul, la probabilité de tous devient nul, danc pour éviter cet cas les termes ayant de probabilité nul on mettons d'une valeur plus petite (tel que : 0 ,001).

En fin on calculant le poids de chaque terme dans le document et la requête puis en calculant la similarité entre la requête et chaque document .selon .

Cas 3 : S'il y a des mots qui sont répété au niveau de la requête dans le module de recherche on utilise le concept probabiliste en suivant les étapes suivantes :

1. Reformulation de la requête;
2. Normalisation des termes de la requête ;
3. Normalisation des termes des documents ;
4. Calculer la probabilité de chaque mot ;

En supposant que ces probabilités soient estimées aussi exactement que possible à partir de toute l'information disponible, aura la meilleur performance possible sur la base de cette information.

Dans ce cas la répétions des termes joue un rôle très important, par ce que les mots très occurrences (sauf les mots vides) souvent devient le plus important par rapport le besoin de l'utilisateur.

Cas 4 : Si la requête est langue supérieure au égale 3 le système va suive le déroulement suivant :

1. Normalisation de requête ;
2. Normalisation de document ;
3. Calculer le poids des termes de chaque document ;
4. Calculer le poids des termes de requête ;
5. Pour calculer le poids on utilisant le « tf*idf » :
   Calculus le « tf » (term frequency):

$$tf_w^d = \left(\frac{x_w^d}{max_{w'}\, x_{w'}^d}\right) \quad (3,7)$$

Calcule idf (inverse document frequency):

$$idf_w = log\,\frac{N}{N_w} \quad (3,8)$$

$$t_w = tf_w^d * idf_w \quad (3,9)$$

6. Calcule la similarité Cosinus

$$cosin(q,d) = \frac{\sum_w t_w^d \ t_w^q}{\sqrt{\sum_w (t_w^d)^2}\ \sqrt{\sum_w (t_w^q\ )^2}} \qquad (3,10)$$

- x_w^q : Nombre occurrences du mot w dans q
- x_w^d : Nombre occurrences du mot w dans le document d
- t_w^d : Version normalisée de x_w^d (poids)
- **N** : Nombre de documents dans la collection
- N_w : Fréquence documentaire de w

Cette figure généralisé les différentes cas précédentes de l'opération de la recherche

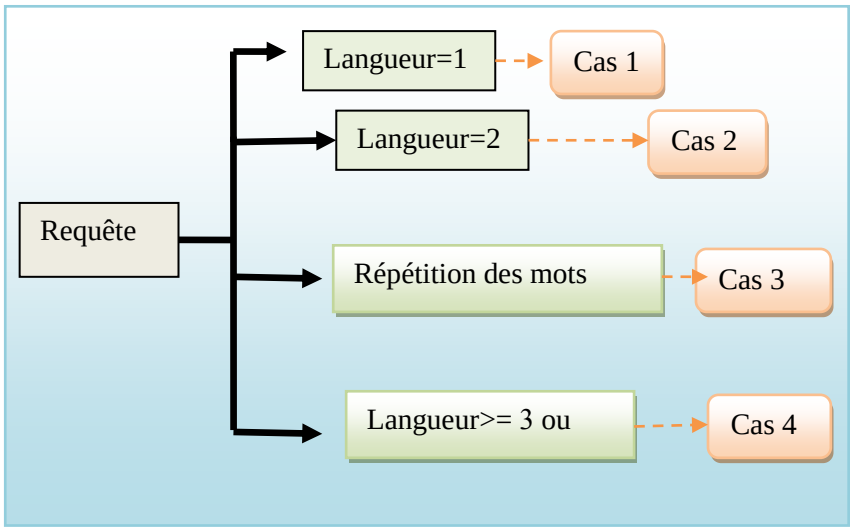


Figure 16 : opération de l'hybridation

5. Conclusion

Dans ce chapitre nous avons présentés la structure générale de notre système IRYSA qui se décompose principalement en deux modules : module d'indexation et module de recherche, ce dernier il consiste plusieurs cas en termes de l'opération de la recherche classé selon des critères tel que la langueur de la requête…etc.

1. Introduction

Dans ce chapitre nous avons essayé de représenter la façon dont notre système a été implémenté, en commençant par la présentation du langage utilisé où on donne les packages qui lui sont associées, puis en passe a l'illustration de l'ensemble des expérimentations que nous avons effectué sur notre système (IRYSA), et qui concernent le calcul de la métrique rappel / précision, et l'évaluation de la qualité des résultats retournés par le système.

2. Environnement de développement

2.1 Présentation du langage

Pour implémenter notre application nous avons utilisés le langage de programmation java, ce dernier est un langage orienté objet interprété de haut niveau, développé par Sun Microsystems. Dont les caractéristiques, sont :

- Le langage java est tout d'abord portable puisqu'un même programme peut être exécuté sur un grand nombre de systèmes d'exploitation ;
- java possède également l'avantage d'être entièrement gratuit tout en proposant la possibilité de pouvoir réaliser des applications commerciales à l'aide de ce langage ;
- La syntaxe de java est beaucoup plus simple, ce qui améliore de façon très significative les temps de développement ;

2.2 packages additionnels

Pour réaliser notre système nous avons eux besoin d'intégrer plusieurs packages, comme SWT[9] (pour développer l'interface graphique) ;en autre packages (voir le tableau8)

Package	Rôle
org.eclipse.swt	Package de base qui contient la définition de constantes et d'exceptions
org.eclipse.swt.custom	Contient des composants particuliers
org.eclipse.swt.dnd	Contient les éléments pour le support du « cliqué / glissé »
org.eclipse.swt.events	Contient les éléments pour la gestion des événements
org.eclipse.swt.layout	Contient les éléments pour la gestion de la présentation
org.eclipse.swt.widgets	Contient les différents composants

Tableau 8: quelques packages qui sont utilisent

[9] Standard Widget Toolkit

2.3 Caractéristiques techniques

Nous avons réalisé notre application dans un environnement décrit par les caractéristiques suivantes :

- Système d'exploitation : Windows 7;
- RAM : 3 GO;
- Résolution de l'écran : 1366 x 768.

2.4 Le corpus de test

Pour démontrer l'intérêt de représenter le contenu textuel par des unités lexicales dans un processus de recherche d'information, nous devons disposer d'un corpus de langue arabe riche en termes. sahih al Bokhari signifie « l'authentique de l'imam Al-Bukhari » est l'un des six grands recueils de hadiths. La plupart des musulmans sunnites le considèrent comme le livre le plus authentique après le Coran. Elle contient 7593 hadiths..

3 L'interface graphique de système IRYSA

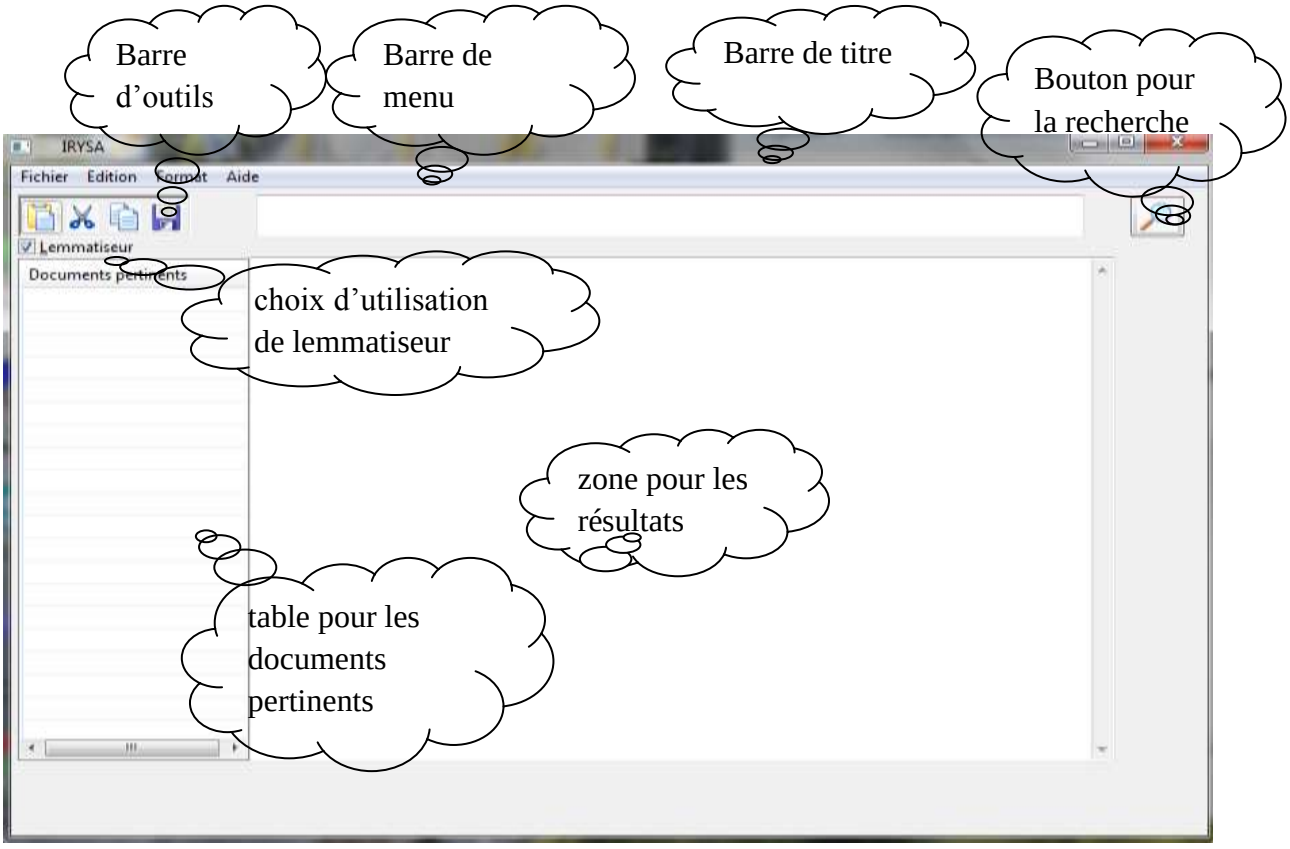


Figure 17 : l'interface graphique de système IRYSA

3.1 Barre de menu

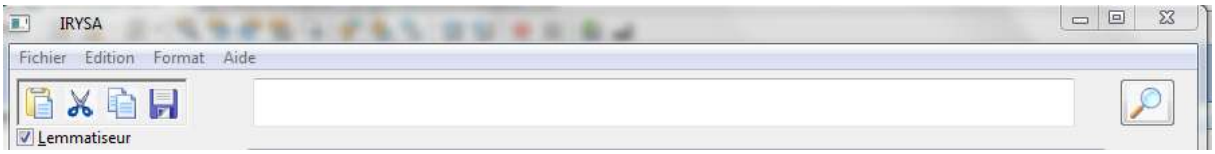


Figure 18: barre de menu

La barre de menu contient :

- 1 : Fichier ;
- 2 : Edition ;
- 3 : Format ;
- 4 : Aide.

3.1.1 Pop up Fichier

Figure 19 : Pop up Fichier

3.1.2 Pop up Edition

Figure 20: Pop up Edition

Le Pop up Edition contient :

- Copier : dupliquer la portion sélectionner ;
- Couper : Enlever la portion sélectionner ;
- Coller : déplacer la portion sélectionner ;
- Sélectionner tout : sélectionner tout le texte.

3.1.3 Pop up Format

Figure 21: Pop up Format

- Police : pour changer la taille et le type du texte.

3.1.4 Pop up Aide

Figure 22: Pop up Aide

Aide : pour le Système.

3.2 Barre des boutons raccourcis

Figure 23: Barre des boutons raccourcis

Cette barre contient les boutons suivants:

- 1: coller ;
- 2 : couper ;
- 3 : copier ;
- 4 : enregistrer le(s) document(s) dans un emplacement quelconque ;

4 Exemple d'affichage d'IRYSA

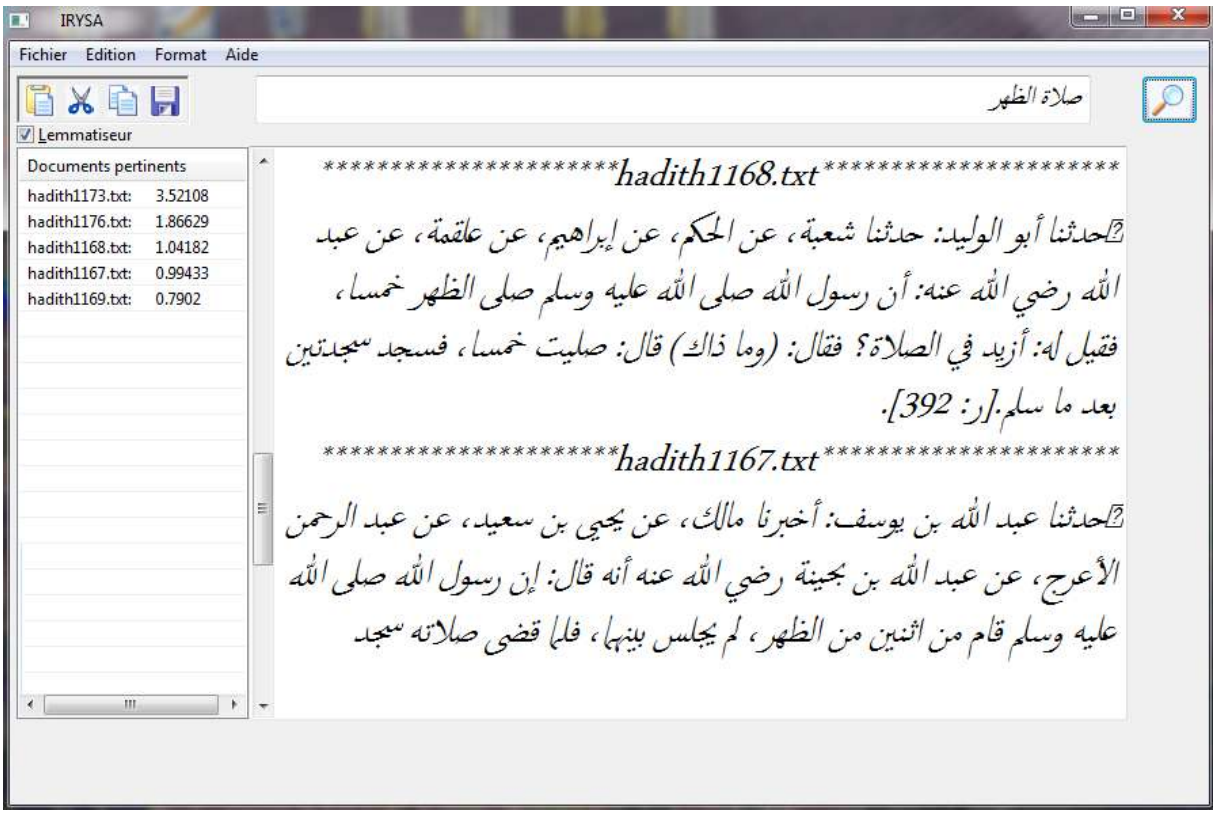


Figure 24: exemple d'affichage de résultat par le système IRYSA

5 Les boites des dialogues du système IRYSA

5.1 Cas 1: le vide

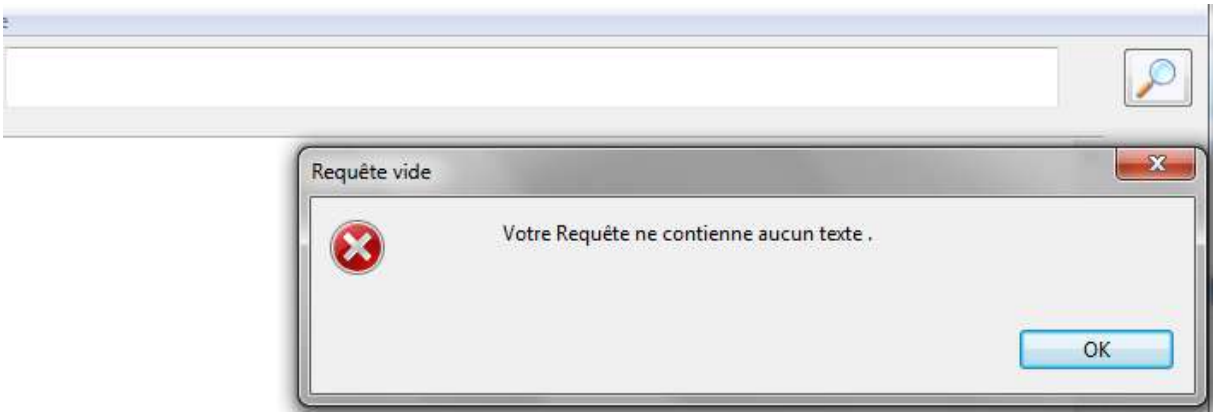


Figure 25: boite de dialogue pour une requête nulle

Cette boite de dialogue s'affiche tant que le champ de la requête est vide.

5.2 Cas 2: Mot outil

Et le boite de dialogue suivant s'affiche quand la requête est un mot outil tel que « على », « الذي », …

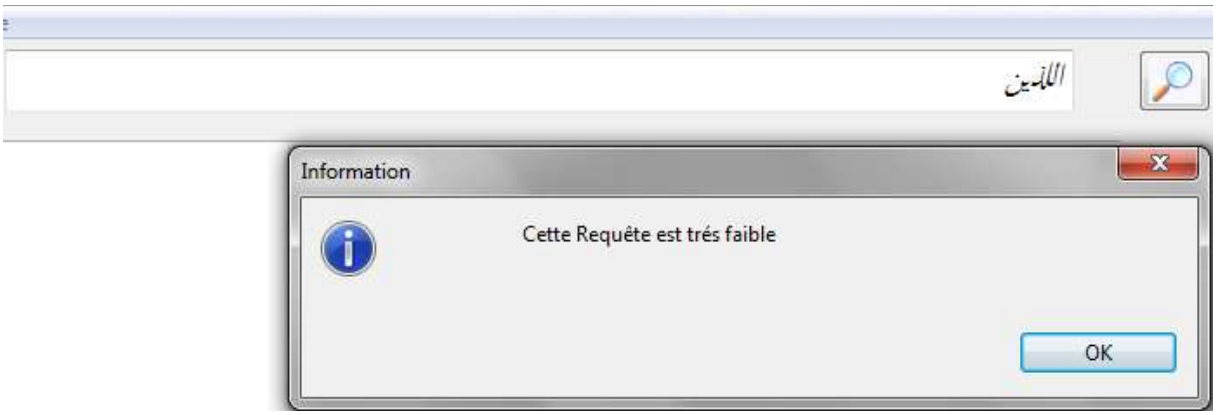


Figure 26: boite de dialogue pour une requête faible

6 Evaluation du système IRYSA

Nous avons sélectionné un corpus de 400 documents, et nous avons effectué 400 requêtes pour établir des tests dans le but de comparer les résultats entre les modèles: booléen, vectoriel, probabiliste, langue et notre système IRYSA. Nous avons trouvé les résultats interprétées par la métrique Rappel (**R**) et Précision (**P**) en prenant en considération quatre (04) cas par rapport à la longueur de la requête .

6.1 Cas 1: longueur de requête = 1 Mot

Requêtes	Booléen		Vectoriel		probabiliste		langue		IRYSA	
الوحي	R	P	R	P	R	P	R	P	R	P
الإيمان	0,40	0,24	0,33	0,24	0,27	0,24	0,40	0,24	0,67	0,59
الجهاد	0,48	0,37	0,44	0,37	0,44	0,37	0,48	0,37	0,44	0,37
الوضوء	0,05	0,04	0,05	0,04	0,05	0,04	0,05	0,04	0,58	0,46
الإسلام	0,50	0,22	0,50	0,22	0,25	0,22	0,50	0,22	0,25	0,22

Tableau 9 : table des tests de (cas 1 : seul mot)

N-B : Pour les figures (de 27 a 51) :

L'axe des x représente l'intervalle de rappel et la précision [0,1], l'axe des y représente le nombre des requêtes.

6.1.1 Pour le modèle Booléen

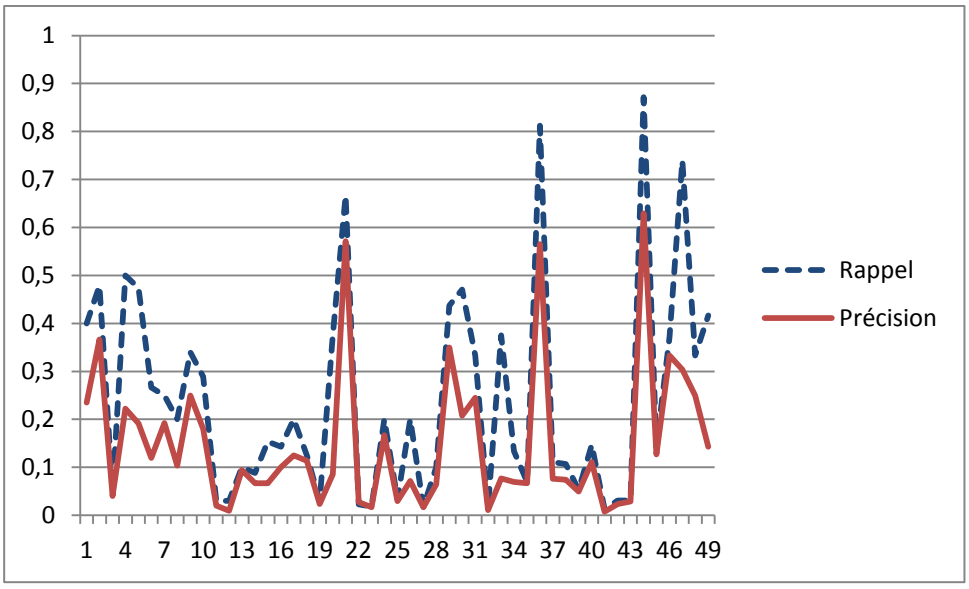


Figure 27 : Courbe rappel-précision du modèle booléen (cas 1: seul mot)

6.1.2 Pour le modèle Vectoriel

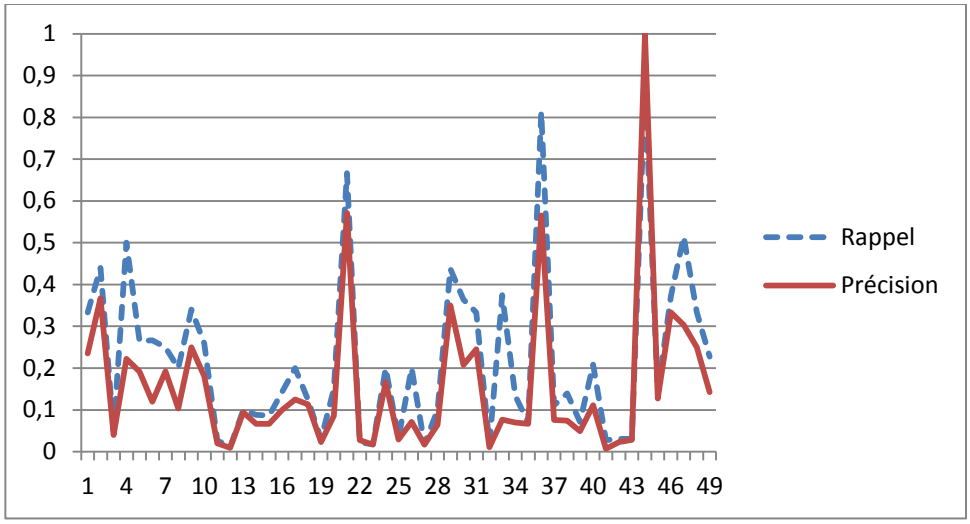


Figure 28 : Courbe rappel-précision du modèle vectoriel (cas 1: seul mot)

6.1.3 Pour le modèle Probabiliste

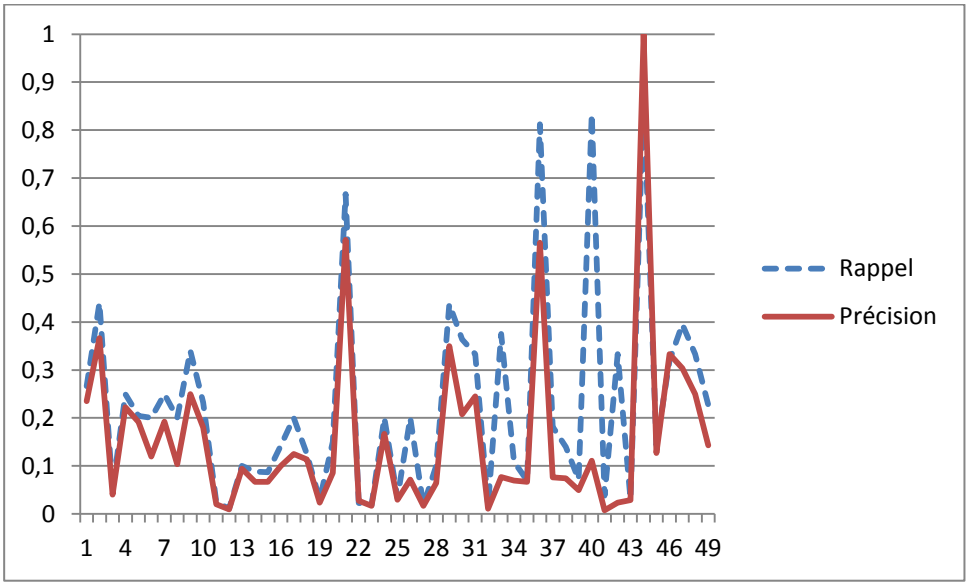


Figure 29: Courbe rappel-précision du modèle probabiliste (cas 1: seul mot)

6.1.4 Pour le modèle de Langue

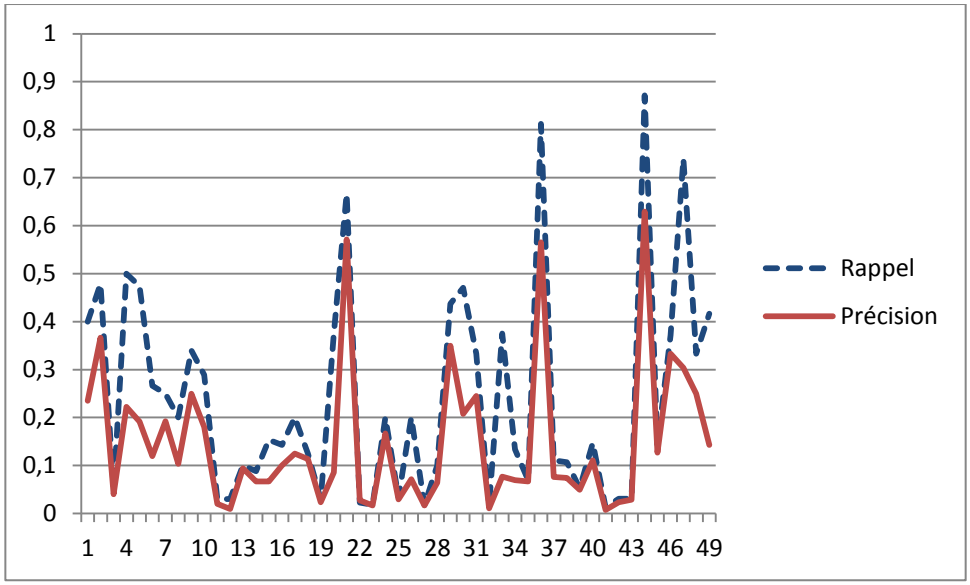


Figure 30 : courbe de rappel-précision du modèle de langue (cas 1: seul mot)

6.1.5 Pour le système IRYSA

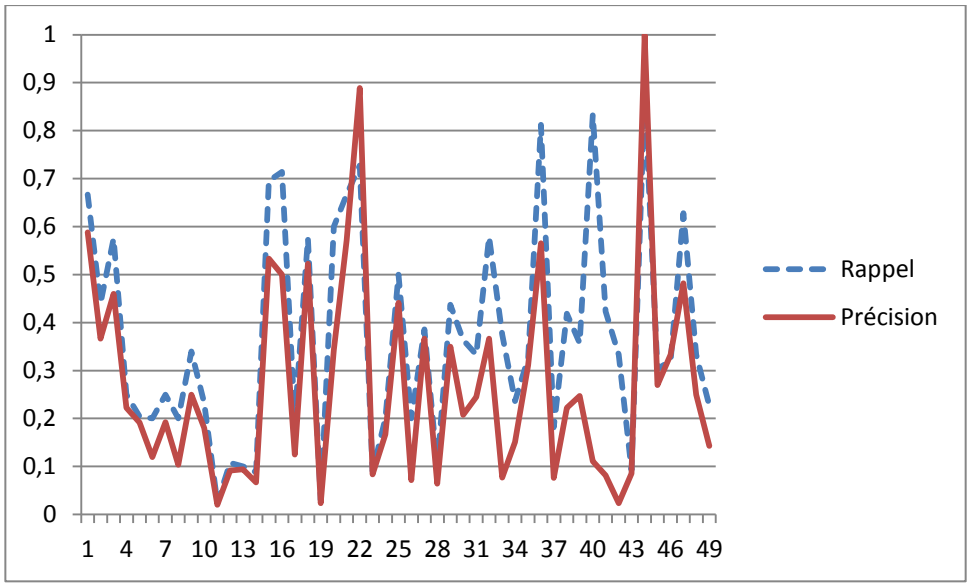


Figure 31 : courbe de rappel-précision du système IRYSA (cas 1: seul mot)

6.1.6 Comparaison entre les cinq modèles (Cas 1: un Seul mot)

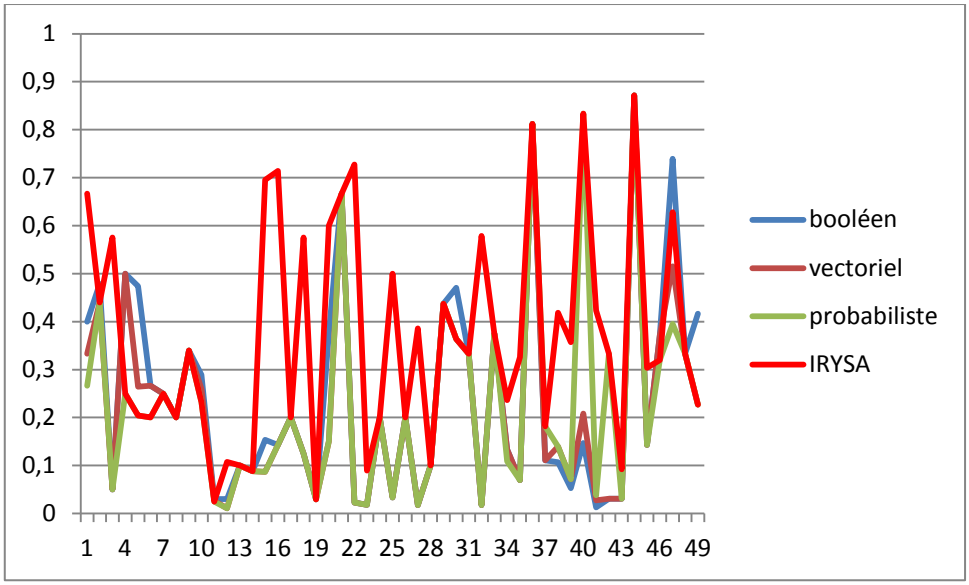


Figure 32 : comparaison entre les rappels (cas 1: seul mot)

Dans la figure on a les courbes sont croissantes et décroissantes à cause de taux des documents pertinentes. L'algorithme de modèle booléen il est donne un résultat acceptable car il utilise le marquage d'un mot (si le mot de requête existe dans document ou non).

On observe que l'algorithme de notre système (IRYSA) est bon et plus garantir par rapport à les autres algorithmes vectoriel et probabiliste car il utilise l'algorithme de modèle booléen et le calcule de la fréquence pour faire l'ordre des documents renvoyé par le système et on ajoutant l'algorithme de lemmatisation pour augmente le nombre des documents pertinentes.

6.2 Cas 2 : longueur de requête=2 mots

Requêtes	booléen		Vectoriel		probabiliste		Langue		IRYSA	
	R	P	R	P	R	P	R	P	R	P
نزول الوحي	0,30	0,01	0,30	0,01	0,30	0,01	0,10	0,00	0,30	0,01
الإسلام الإيمان	0,00	0,00	0,20	0,01	0,20	0,01	0,00	0,00	0,20	0,01
الإيمان الإسلام	0,50	0,03	0,80	0,04	0,80	0,04	0,25	0,01	0,80	0,04
لايؤمن أحدكم	0,00	0,00	0,95	0,05	0,95	0,05	0,05	0,00	0,95	0,05
إيمان بالله	0,00	0,00	0,50	0,03	0,50	0,03	0,05	0,00	0,50	0,03

Tableau 10 : table des tests (cas 2: deux mot)

6.2.1Modèle booléen

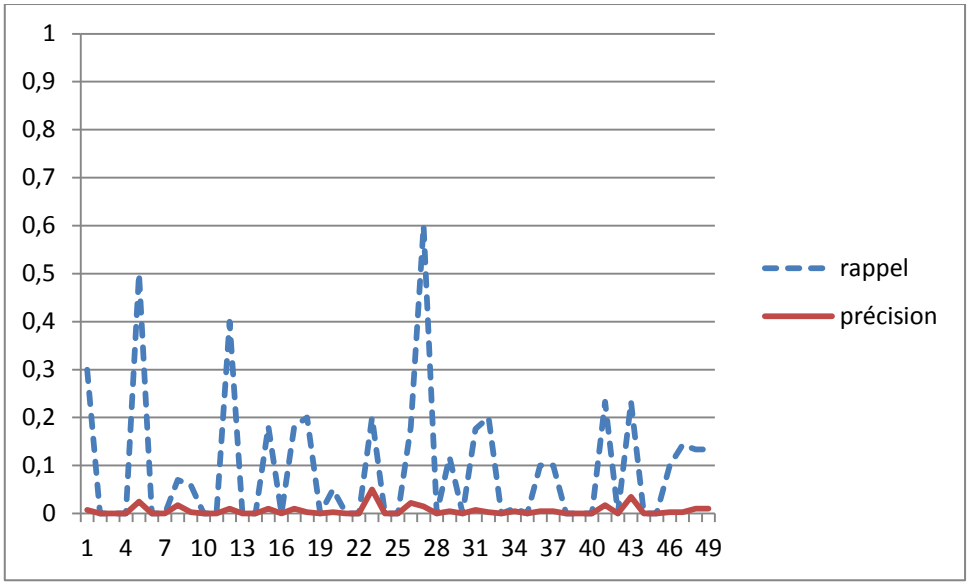


Figure 33 : courbe rappel-précision modèle booléen cas 2

(longueur de requête égal 2 mots)

6.2.2 Pour modèle vectoriel

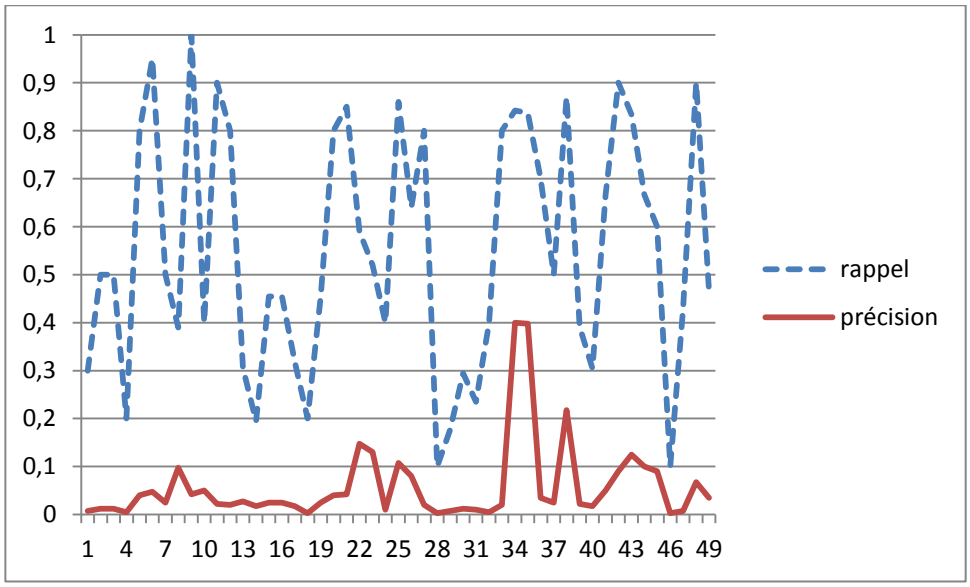


Figure 34: courbe rappel-précision modèle vectoriel cas2
(longueur de requête égal 2 mots)

6.2.3 Pour modèle probabiliste

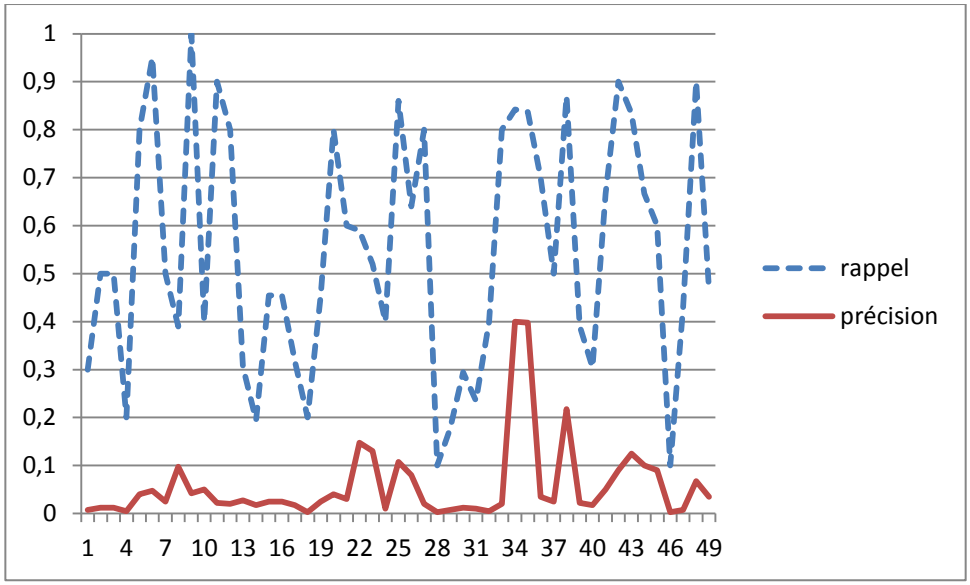


Figure35 : courbe rappel-précision modèle probabiliste cas2 (longueur de requête égal 2 mots)

6.2.4 Pour modèle de langue

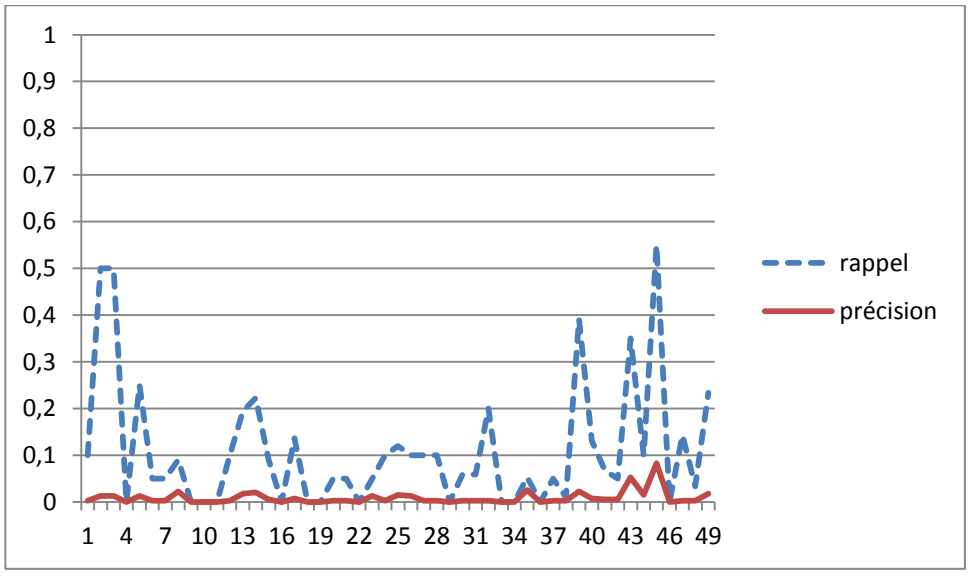


Figure 36: courbe rappel-précision modèle de langue cas2 (longueur de requête égal 2 mots)

6.2.5 Pour IRYSA

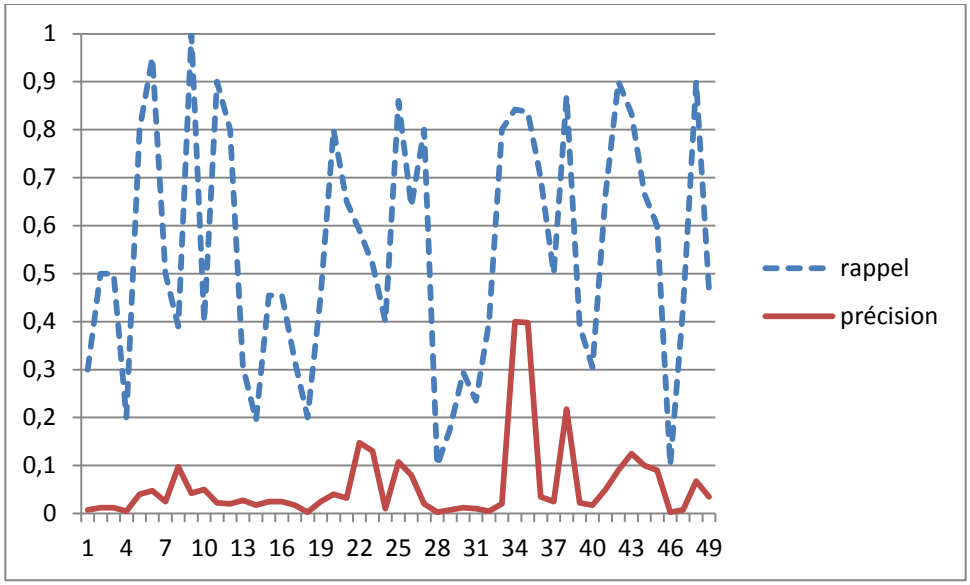


Figure 37: courbe rappel-précision système IRYSA cas2 (longueur de requête égal 2 mots)

6.2.6Comparaison cas 2 (longueur de requête égal 2 mots)

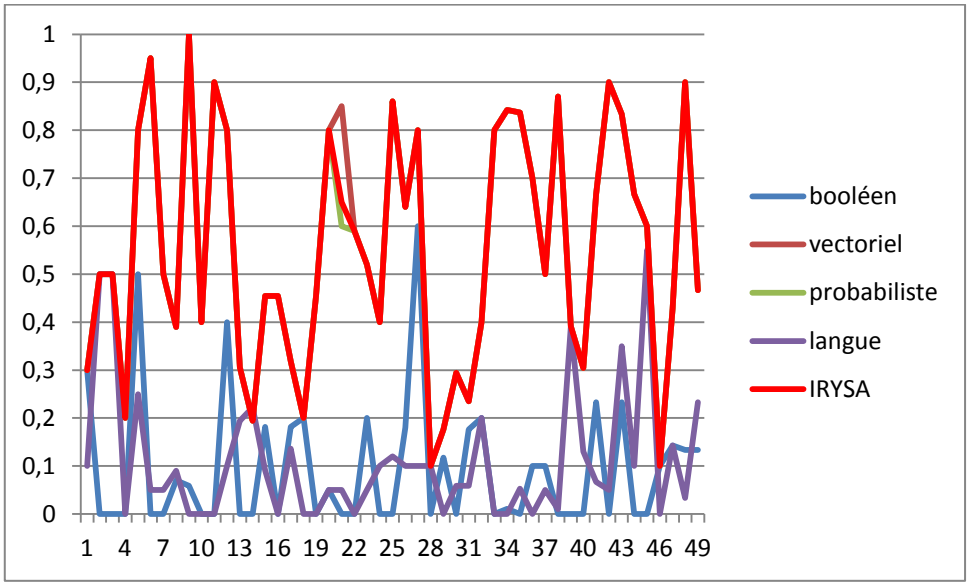


Figure 38: courbes des rappels en cas 2(longueur de requête égal 2 mots)

D'après la figure on n'observe que l'algorithme de notre système (IRYSA) donne un bon résultat par rapport un autre algorithme de modèle booléen et probabiliste car il utilise la notion de n-gramme lorsque le mot non nul (le mot un et appartienne a le document et le mot deux aussi appartienne) en combinaison a l'algorithme de lemmatisation.

6.3 Pour cas 3 (des mots de la même racine)

requêtes	Booléen		Vectoriel		probabiliste		Langue		IRYSA	
	R	P	R	P	R	P	R	P	R	P
سلم المسلمون	0,40	0 ,40	0,90	1,00	0,25	0,90	0,86	0,00	0,01	0,00
المهاجر من هجر ما نهى الله	0,01	0,01	0,40	0,13	0,90	0,86	0,01	0,01	0,92	0,67
يحب لأخيه ما يحب لنفسه	0,00	0,00	0,80	0,67	0,95	0,76	0,00	0,00	0,88	0,70
آية الإيمان وآية النفاق	0,00	0,00	0,96	0,90	1,00	0,83	0,00	0,00	0,96	0,84
كفر بعد كفر	0,00	0,00	0,15	0,13	0,25	0,22	0,00	0,00	0,38	0,33

Tableau 11 : table des tests (cas 3 : la requête a des mots de même racine)

6.3.1 Pour modèle booléen

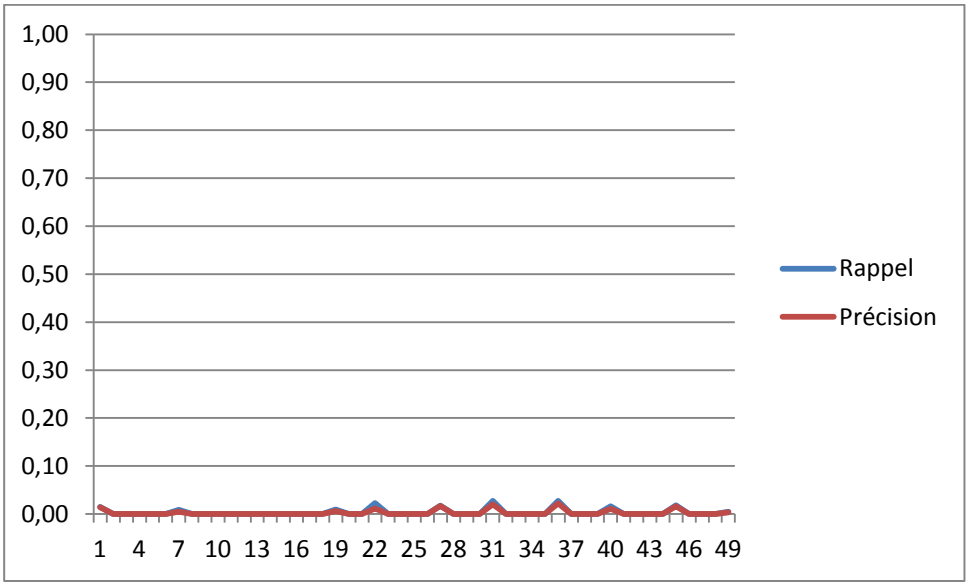


Figure 39: courbe rappel-précision modèle booléen cas 3

(la requête a des mots de même racine)

6.3.2 Pour modèle vectoriel

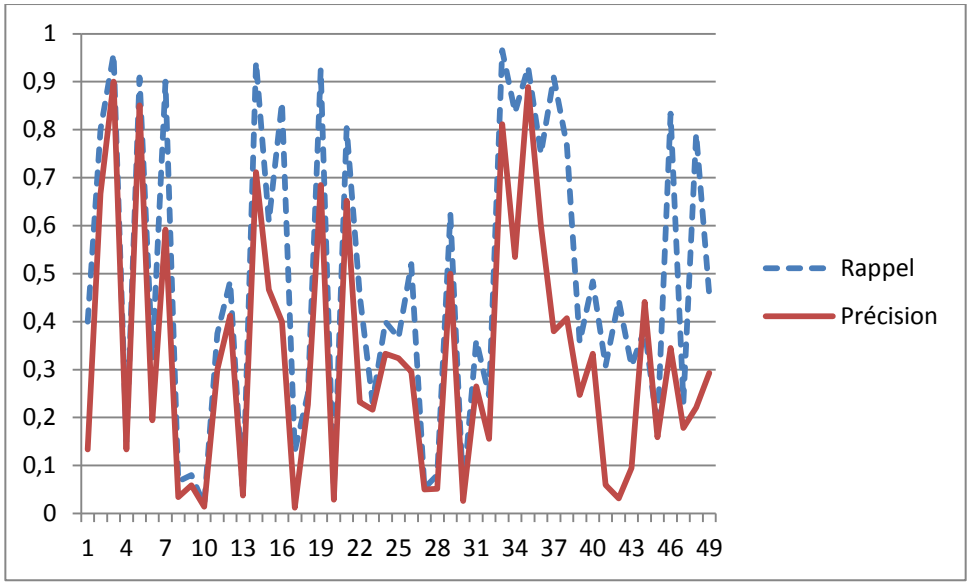


Figure 40: courbe rappel-précision modèle vectoriel cas 3(la requête a des mots de même racine)

6.3.3 Pour modèle probabiliste

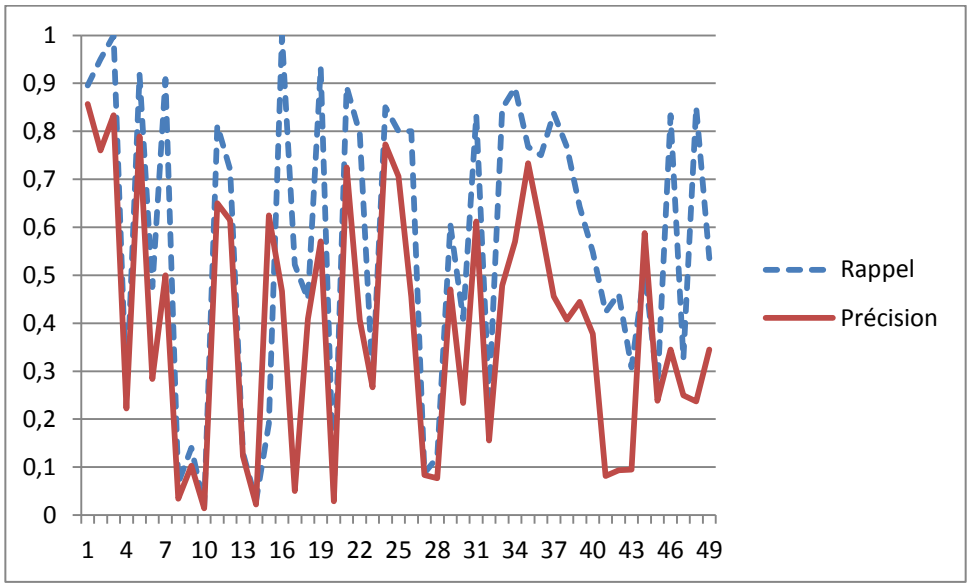


Figure 41: courbe rappel-précision modèle probabiliste cas 3(la requête a des mots de même racine)

6.3.4 Pour le modèle de langue

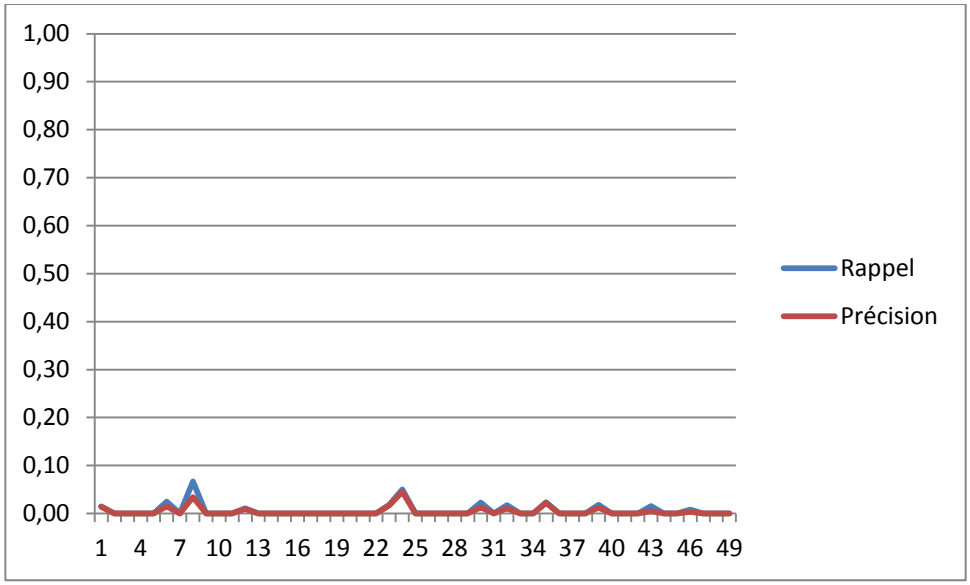


Figure 42: courbe rappel-précision modèle langue cas 3

(La requête a des mots de même racine)

6.3.5 Pour IRYSA

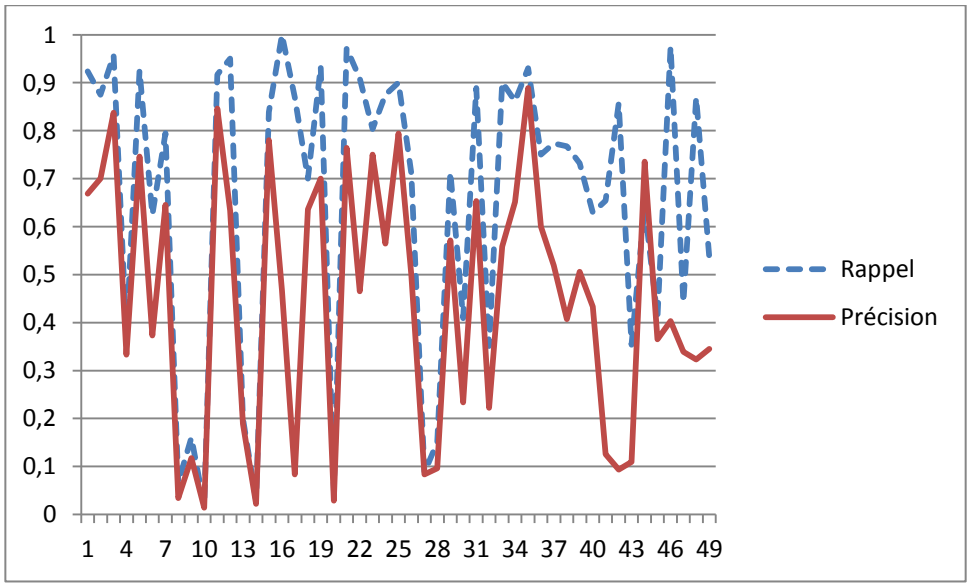


Figure 43: courbe rappel-précision système IRYSA cas 3

(La requête a des mots de même racine)

6.3.6 Comparaison cas 3(la requête a des mots de même racine)

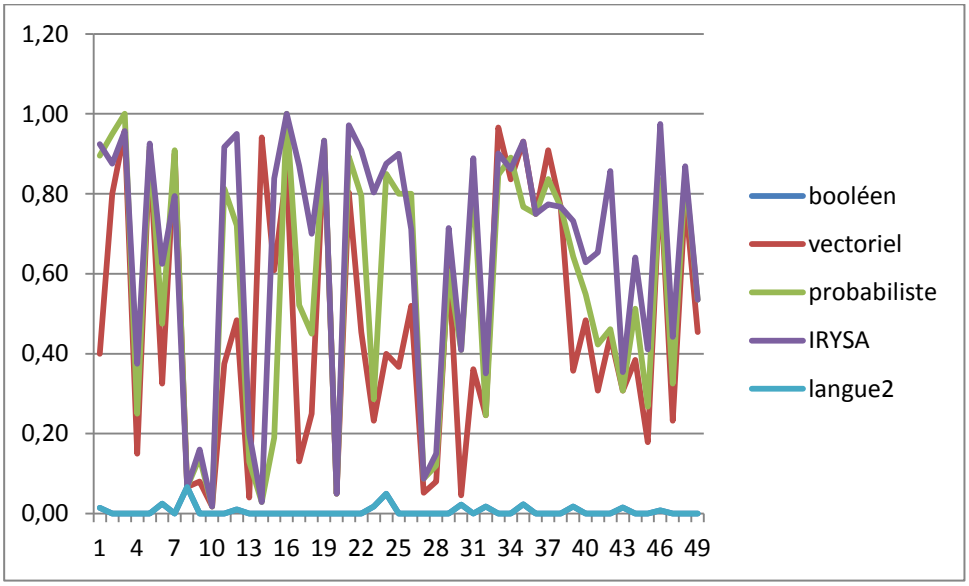


Figure 44:comparaison courbes des rappels cas 3

(La requête a des mots de même racine)

D'après la figure en vu que l'algorithme de notre système (IRYSA) plus précise lorsque les mots sont répètes dans les requêtes (mots même racine) par rapport les autres algorithmes et le taux des documents pertinentes renvoyer par notre système (IRYSA) est plus grand car notre système (IRYSA) utilise la notion de probabilité toujours avec l'algorithme de la lemmatisation par contre les autres algorithmes.

6.4 Pour cas 4(longueur de requête supérieur ou égal 3)

Requêtes	Booléen		Vectoriel		probabiliste		langue		IRYSA	
	R	P	R	P	R	P	R	P	R	P
الفرق بين صلاة التطوع و التهجد	0,02	0,00	0,40	0,05	0,58	0,07	0,10	0,01	0,58	0,07
غير أنه يتم الركوع والسجود	0,06	0,01	0,22	0,03	0,18	0,02	0,02	0,00	0,26	0,03
متى يحل وقت صلاة التهجد	0,06	0,01	0,52	0,07	0,40	0,05	0,34	0,04	0,52	0,07
ارتحل قبل أن تزيغ الشمس	0,00	0,00	0,32	0,04	0,60	0,08	0,10	0,01	0,96	0,12
حتى يقوم من جوف الليل	0,06	0,01	0,98	0,12	0,68	0,09	0,06	0,01	0,98	0,12

Tableau 12 : table des tests (cas 4 : longueur de requête supérieur ou égal 3)

6.4.1 Pour modèle booléen

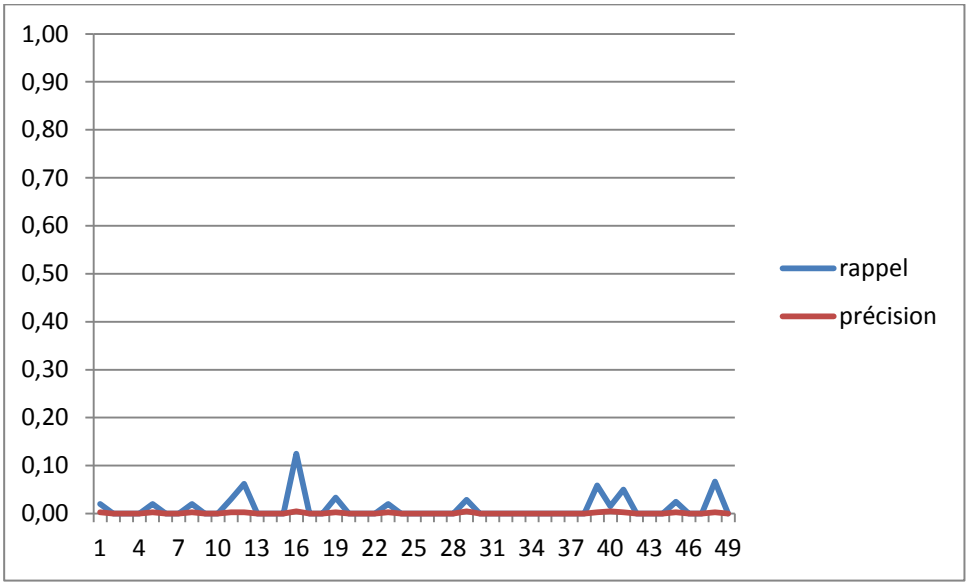


Figure 45: courbe rappel-précision modèle booléen cas 4(longueur de requête supérieur ou égal 3)

6.4.2 Pour modèle vectoriel

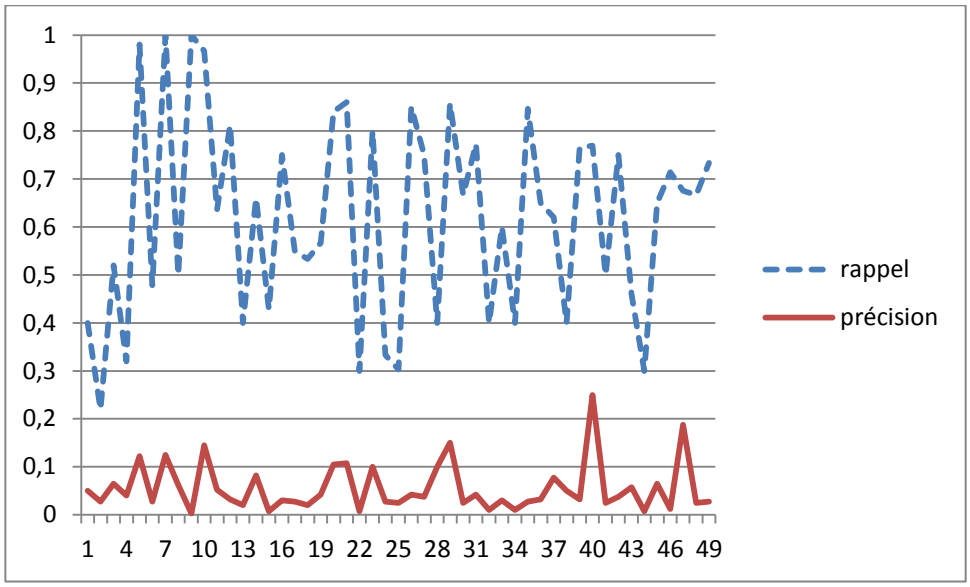


Figure 46: courbe rappel-précision modèle vectoriel cas 4(longueur de requête supérieur ou égal 3)

6.4.3 Pour modèle probabiliste

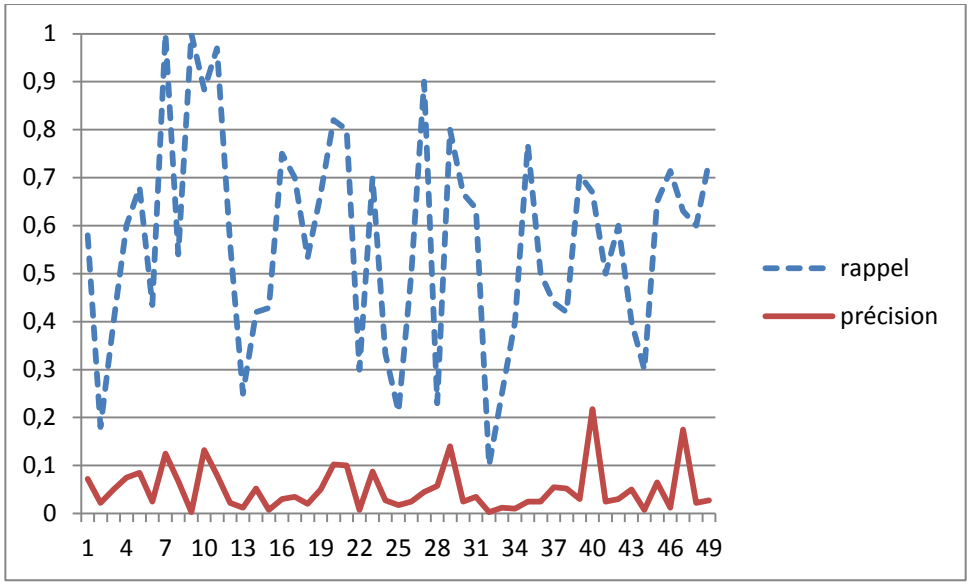


Figure 47: courbe rappel-précision modèle probabiliste cas 4(longueur de requête supérieur ou égal 3)

6.4.4 Pour le modèle de langue

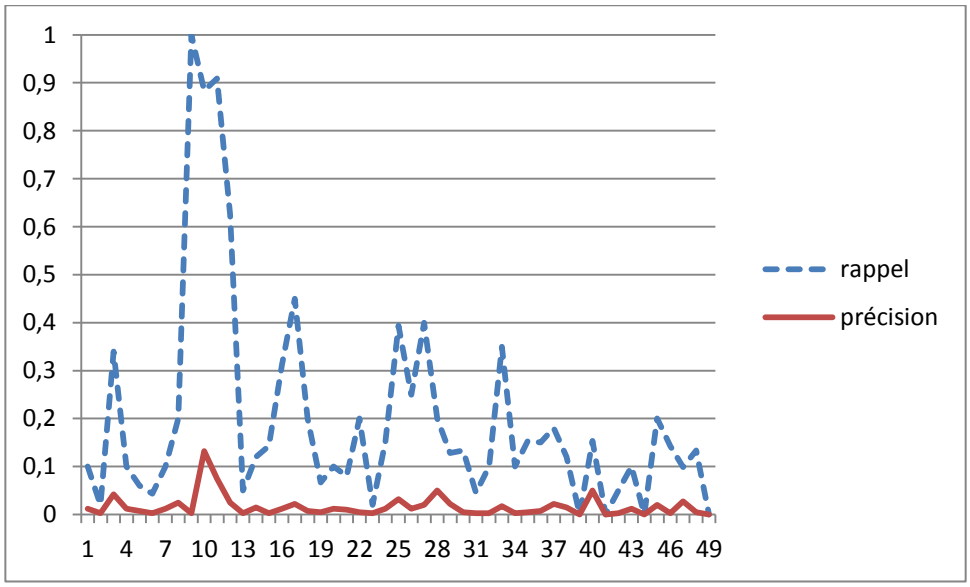


Figure 48: courbe rappel-précision modèle de langue cas 4(longueur de requête supérieur ou égal 3)

6.4.5 Pour système IRYSA

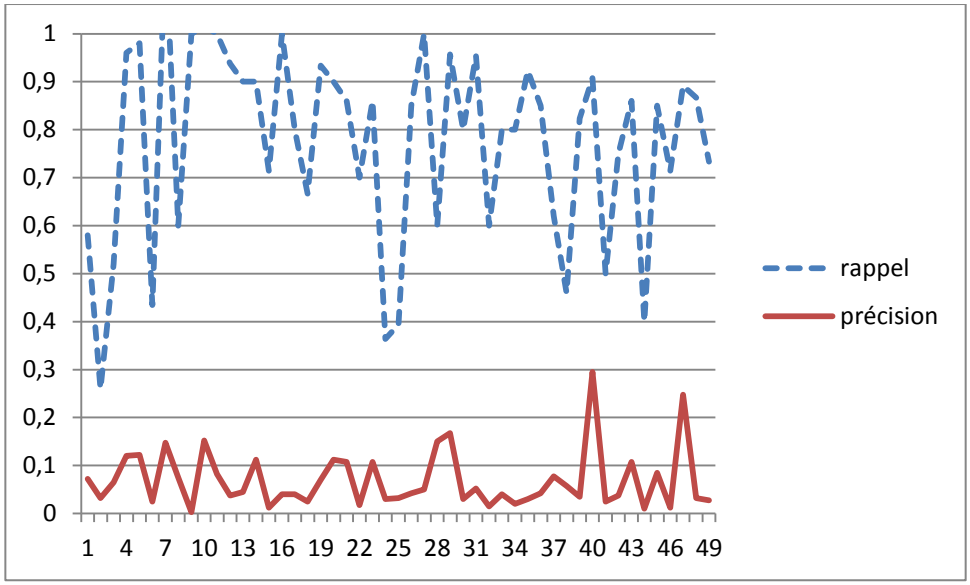


Figure 49: courbe rappel-précision système IRYSA cas 4(longueur de requête supérieur ou égal 3)

6.4.6Comparaison cas 4(longueur de requête supérieur ou égal 3)

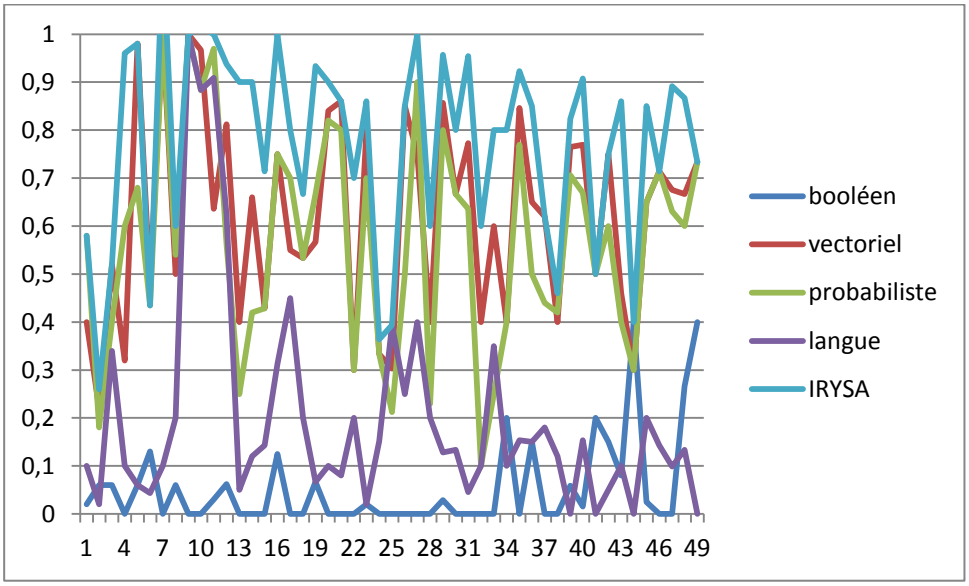


Figure 50 : comparaison courbes des rappels de cas 4(longueur de requête supérieur ou égal 3)

D'après la figure nous observons l'algorithme de notre système (IRYSA) est très bon par rapport à les autres algorithmes car notre système (IRYSA) prend l'algorithme de modèle vectoriel et en ajoute les calcules de fréquences de chaque mot en combinaison avec l'algorithme de lemmatisation.

6.5 Rappel IRYSA avec et sans lemmatiseur

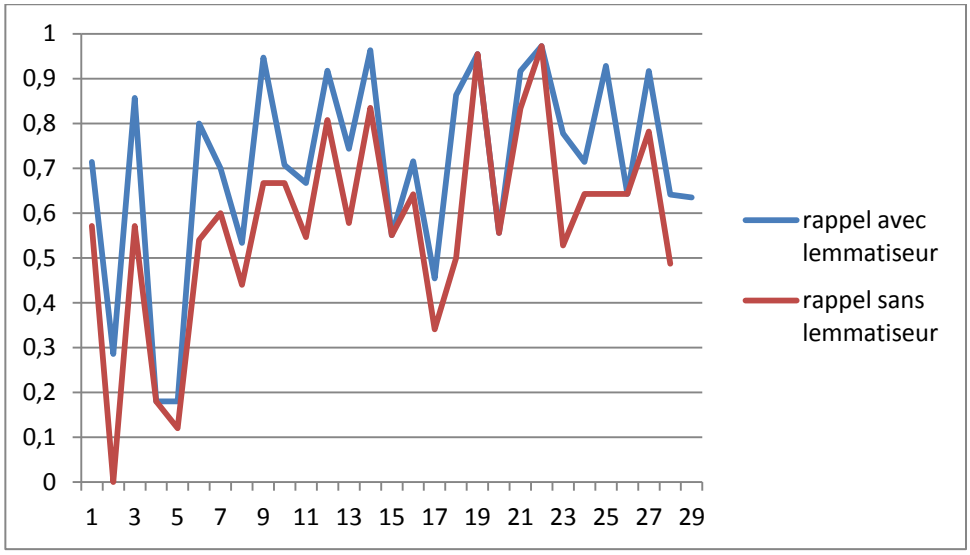


Figure 51: courbe de rappel IRYSA avec et sans lemmatiseur

D'après la figure(51) en conclure que les résultats de système IRYSA avec lemmatiseur est très bon par rapport les résultats sans lemmatiseur.

7 Conclusion

Dans ce dernier chapitre nous avons donné une description de l'interface graphique que nous avons implémentée afin de simplifier la tache de l'utilisation de notre système de recherche, et le langage de programmation que nous avons utilisé. Puis nous sommes passés à la partie la plus importante de notre travail, où nous avons représenté l'ensemble des expérimentations que nous avons effectué, dans le but d'évaluer la qualité et la fiabilité de notre système de Recherche IRYSA.

Au niveau de ce mémoire, on a présenté notre travail qui concerne la conception et la réalisation d'un système de recherche d'informations en utilise le lemmatiseur dans le champ de la langue arabe.

Notre projet est de créer un système recherche d'après les modèles existante, il se compose de deux modules. Un module d'indexation qui permet l'extraction des termes représentatifs de chaque document du corpus. Et pour avoir une bonne représentation de ces documents on a besoin de passer par un ensemble d'étapes qui concerne l'extraction des unités lexicales, l'élimination des mots vide, la lemmatisation de termes d'index, et la génération de l'index du corpus.

Le deuxième module de notre système est le module de recherche, qui prend en entrée la requête utilisateur, il traite de la même manière que les documents du corpus.

Un système de recherche d'information doit représenter, stocker et organiser l'information puis fournir à l'utilisateur les éléments correspondant au besoin exprimé par sa requête.

Ainsi un SRI en langue arabe doit prendre en considération ses caractéristiques singulières et proposer des outils et des techniques automatiques afin de permettre son traitement informatique.

Les perspectives de nos travaux sont multiples :

- ✓ Enrichir de la base en ajout d'autre corpus ;
- ✓ Utilisation de « word net » par exemple pour améliorée la recherche avec des synonymes et pour donne un meilleur résultat ;

Des testes cas1 : un mot

Requêtes	booléen	vectoriel	probabiliste	langue	IRYSA
الوحي	5	4	4	4	10
الإيمان	11	6	4	11	13
الجهاد	2	1	1	2	3
الوضوء	10	7	3	10	15
الإسلام	9	5	7	9	11
العلم	8	8	8	8	8
الإحسان	5	1	4	2	6
جبريل	3	1	1	1	6
الصلاة	10	7	7	10	17
الرسول	11	3	9	10	13
التيمم	2	2	2	1	2
صعيدا	1	1	1	1	1
ماء	55	11	10	55	55
الإسراء	3	1	2	3	3
سدرة	2	2	2	2	5
المسك	10	1	1	4	10
عيسى	5	2	1	4	7
الثوب	5	3	2	3	7
جلباب	12	1	1	1	12
إزار	3	2	1	2	3
الاشتمال	6	2	4	1	7
السري	1	0	0	0	1
الصبيان	17	1	1	10	20
الإداوة	2	1	1	1	2
القميص	1	1	1	1	1
متقين	1	1	1	1	1
الثلج	1	1	1	1	1
كبر	10	5	2	8	13
الإمام	7	3	2	5	10
السجود	14	10	10	11	16
المسلم	11	6	8	10	14
أقاتل	10	1	1	6	15
دم	3	2	3	2	3
القبلة	6	3	2	6	6
المشرق	3	1	0	1	3
المقام	10	9	9	10	13
الصفا	11	3	1	5	6
المروة	6	2	6	5	10
العمرة	7	3	4	2	7
الكعبة	8	5	7	7	10
المقدس	10	1	1	6	10
صراط	2	1	1	2	3
السفهاء	12	2	2	5	15
العصر	30	15	15	20	34
الفريضة	5	3	1	4	8
المساجد	13	7	10	10	16
الركوع	17	3	5	7	22

الوداع	2	0	1	2	2
الخيل	4	3	3	2	5
التيمن	11	2	4	2	13
الصور	3	3	3	3	3
الخلق	7	4	5	3	10
المقابر	12	6	10	6	12
النار	14	7	10	9	14
اليهود	5	3	3	5	5
الأرض	9	6	3	5	11
الأنبياء	15	12	12	12	15
النصارى	7	4	5	7	13
الغنائم	3	3	3	3	3
الشفاعة	4	2	1	4	4
الفقراء	6	3	2	1	6
الملائكة	13	10	8	11	15
الحدث	5	5	5	5	5
المطر	10	1	1	8	17
الزكاة	22	12	15	20	22
الجنة	20	10	12	15	20
المنبر	13	3	10	13	15
الحراب	30	12	14	20	35
البيع	6	2	3	4	6
الشراء	12	6	8	12	14
التقاضي	15	12	10	9	17
الخمر	5	3	5	2	5
الاغتسال	10	9	5	9	12
الدنيا	12	5	8	10	14
المشرك	13	5	5	9	15
السيئات	6	4	4	4	14
البيعة	2	2	2	2	4
الصوم	9	3	2	5	9
الصدقة	8	4	3	5	11
الحسنات	8	2	1	10	12
جهنم	5	2	3	2	5
الظهر	10	2	4	6	10
الخير	2	1	1	1	3
الشمس	40	12	16	23	50
الفجر	13	5	2	10	14
العتمة	2	1	1	1	2
المنافقين	6	3	2	5	6
الفوائت	3	1	2	2	3
الأذان	7	2	2	5	10
الإقامة	4	1	1	3	4
الدعاء	7	4	3	5	8
القيامة	13	3	3	8	13
الجماعة	15	5	6	11	19
التهجير	11	2	3	5	10
الشهيد	21	2	2	12	24
المريض	3	1	1	1	3
الإمامة	14	4	2	5	16

الخشوع	15	3	2	5	17
البصر	12	2	4	10	15
الغضب	17	4	6	11	23

Des testes cas2 :deux mots

Requetes	booléen	vectoriel	probabiliste	langue	IRYSA
نزول الوحي	3	3	3	1	3
النبي الوحي	0	5	5	5	5
جبريل الوحي	0	5	5	5	5
الإسلام الإيمان	0	2	2	0	2
الإيمان الإسلام	10	16	16	5	16
لا يؤمن أحدكم	0	19	19	1	19
إيمان بالله	0	10	10	1	10
ذرة من خير	7	39	39	9	39
متى يزيد الإيمان	1	17	17	0	17
إذا أسلم العبد	0	20	20	0	20
الإيمان والإسلام	0	9	9	0	9
علمه الكتاب	4	8	8	1	8
الإسلام و العلم	0	11	11	7	11
أدب العلم	0	7	7	8	7
المسلم و المؤمن	4	10	10	2	10
العلم و العمل	0	10	10	0	10
و العلم الجهل	4	7	7	3	7
الوضوء الصحيح	1	1	1	0	1
موجبات الوضوء	0	10	10	0	10
الوضوء الماء	1	16	16	1	16
إداوة من ماء	0	63	63	1	13
التيمن في الوضوء	0	59	59	0	59
التيمم في الإسلام	20	52	52	5	52
الصحيح الغسل	0	4	4	1	4
وضوءه للصلاة	0	43	43	6	43
سبحان الله	9	32	32	5	32
الحيض و الغسل	6	8	8	1	8
الوجه والكفين	0	1	1	1	1
التيمم الوجه	2	3	3	0	3
يعني تيمم	0	5	5	1	5
عليك بالصعيد	3	4	4	1	4
أدركته الصلاة	1	2	2	1	2
شعبة بيديه	0	8	8	0	8
صلى في ثوب	2	160	160	0	160
يصلي أحدكم	0	159	159	10	159
صلاة الظهر	2	14	14		14
صلاة العصر	2	10	10	1	10
فأمره أن يؤذن	0	87	87	1	87
صلاة الجماعة	0	9	9	9	9
يناجي ربه	0	7	7	3	7
صل ركعتين	7	20	20	2	20
صعد المنبر	0	36	36	2	36

صلى على قبره	14	50	50	21	50
المسجد فليركع	0	40	40	6	40
أراد أن يؤذن	0	36	36	3	36
وعليكم بالسكينة	1	1	1	0	1
فناد بالصلاة	1	3	3	1	3
حي على الصلاة	4	27	27	1	27
يشفع الأذان	4	14	14	7	14
صلاة الجمعة	13	45	45	1	45
تطهرتم ليومكم	1	1	1	1	1
يوم الجمعة	4	47	47	22	47
أذن المؤذن	1	11	11	5	11
خطبة الجمعة	0	41	41	15	41
نبكر إلى الجمعة	4	54	54	1	54
فصل ركعتين	0	8	8	3	8
صلاة العيد	2	4	4	2	4
ويكبر المكبر	7	2	2	1	2
جماعة المسلمين	0	1	1	1	1
يصلون العيدين	1	3	3	1	3
أيام عيد	8	8	8	6	8
قبل الصلاة	8	15	15	6	15
صلاة الليل	8	7	7	4	7
استقبل القبلة	0	7	7	2	7
وتره إلى السحر	5	18	18	1	18
بشق المسافر	5	8	8	1	8
وتظهر الفتن	0	4	4	1	4
فصلى ركعتين	5	12	12	5	12
كسوف الشمس	0	25	25	20	25
صلاة الكسوف	3	1	1	1	1
الشمس والقمر	2	25	25	19	25
تجلت الشمس	0	24	24	11	27
صلاة الفجر	0	1	1	1	1
وسجد الناس	1	12	12	8	12
القوم إلا سجد	13	13	13	5	13
اركان الإيمان	6	50	47	20	53
فضل العلم	6	50	38	10	50
موسى والخضر	16	98	76	23	100
تقصير الصلاة	14	76	76	13	81
وضوء الرسول	11	67	67	61	105
سحاب السماء	0	45	56	34	60
فرض الوضوء	1	60	45	12	78
تيمم الرسول	11	22	22	5	20
صلاة الخوف	4	57	41	4	38
صلاة الفجر	1	20	19	6	42
الآذان والصلاة	6	16	16	2	20
خسفت الشمس	4	19	27	8	33
صلاة الجمعة	11	18	20	9	35
الصلاة في العيد	6	18	18	1	28
ايتاء الزكاة	6	70	70		77
زكاة الفطر	18	27	27	6	30
اتباع الجنائز	11	10	8	1	10

الدواب المحرم	8	17	17	2	20
رمضان صائم	8	30	20	7	37
صلاة الضحى	2	11	17	3	23
اجر الفعل	11	11	19	2	41
المسيح الدجال	15	43	55	9	59
طواف الافاضة	7	19	17	1	22
حج البيت	17	28	26	0	37
وقت الصلاة	2	75	59	3	20

Teste de 4 cas

Requetes	booléen	vectoriel	probabiliste	Langue	IRYSA
الفرق بين صلاة التطوع و التهجد	1	20	29	0	29
غير أنه يتم الركوع والسجود	3	11	9	1	13
متى يحل وقت صلاة التهجد	3	26	20	0	26
ارتحل قبل أن تزيغ الشمس	0	16	30	2	48
حتى يقوم من جوف الليل	3	49	34	3	49
إنما جعل الإمام ليؤتم به	3	11	10	1	10
إذا قام للتهجد من الليل	0	50	50	5	59
الذي يثلغ رأسه بالحجر	3	25	27	10	30
أفلا أكون عبدا شكورا	0	1	1	1	1
آمن ما كان، بمنى ركعتين	0	58	53	53	61
ما رأيته صلى غير ذلك اليوم	1	21	32	30	33
بين بيتي ومنبري روضة من رياض الجنة	1	13	9	10	15
مسجد الحرام، ومسجد الأقصى	0	8	5	1	18
صلى ثماني ركعات	0	33	21	6	45
لا تتحروا طلوع الشمس ولا غروبها	0	3	3	1	5
لا يدع أربعا قبل الظهر	2	12	12	5	16
إذا جاء أحدكم والإمام يخطب	0	11	14	9	16
نسي الفاتحة سهو	0	8	8	3	10
إنما منعني أن أرد عليك أني كنت أصلي	2	17	20	2	28
حتى يظل الرجل إن يدري كم صلى	0	42	41	5	45
صلى وراءه قوم قياما	0	43	40	4	43
رب اغفر لي	0	3	3	2	7
نهي أن يصلي الرجل مختصرا	1	40	35	1	43
فتلقاها منه الناس	0	11	11	5	12
من سلم المسلمون من لسانه ويده	0	10	7	13	13
عصموا مني دماءهم وأموالهم	0	17	10	5	17
وسلم الظهر ركعتين، والعصر ركعتين	0	15	18	8	20
النبي صلى الله عليه وسلم يمسح على الخفين	0	40	23	20	60
كنت أغتسل أنا والنبي صلى الله عليه وسلم من إناء واحد	2	60	56	9	67
إذا لم يجد الماء لا يصلي	0	10	10	2	12
(فقال: (صل ركعتين	0	17	14	1	21
مري غلامك النجار	0	4	1	1	6
يا أهل الخيمة	0	12	5	7	16
والذي نفس محمد بيده	2	4	4	1	8

دخول وقت زكاة الفطر	0	11	10	2	12
كيف نزل الوحي على الرسول	3	13	10	3	17
ليلة القدر جوف الليل	0	31	22	9	31
السلم كيل معلوم	0	20	21	6	23
صيام رمضان وقيام الليل	1	13	12	0	14
الحياء شعبة من الإيمان	2	100	87	20	118
من قام رمضان ايمانا واحتسابا	4	10	10	0	10
هل الإيمان هو الاسلام نفسه	3	15	12	1	15
ما هي موجبات الوضوء	4	23	20	5	43
متى فرض الوضوء	4	3	3		4
كيف تيمم الرسول	1	26	26	8	34
صلاة الجماعة وصلاة الفرد	0	5	5	1	5
ايتاء الزكاة والصدقة	0	75	70	11	99
حجة الوداع ووصايا الرسول	4	10	9	2	13
الطواف و السعي بين الصفا والمروى	6	11	11		11
الربا و الصدقات الصحيحة	1	30	33	18	75
يعتكف في رمضان و يقوم اليل	0	87	77	7	101
مكة الصفا و المروة	0	90	50	7	98
غسل و تكفين الميت	0	28	20	1	31
لا يؤمن أحدكم حتى أكون أحب إليه	0	97	97	0	102
يكره أن يلقى في النار	0	88	86	0	110
يؤذن بالصلاة يوم الفطر	1	50	44	10	56
العنزة بين يديه تحمل	5	75	70	0	78
أمرهن أن يتصدقن، فجعلت المرأة تلقي	5	93	93	0	93
المصدق عشرين درهما أو شاتين	0	42	32	0	46
خمس صلوات في يومهم وليلتهم	5	67	80	0	72
كانت مستقبلة المسجد	0	40	39	4	49
ليس على المسلم في فرسه وغلامه صدقة	0	58	58	1	67
فكبر في كل سجدة وهو جالس	0	87	71	0	105
لو كنت ثم لأريتكم قبره	0	55	43	1	62
كان يجمع بين الرجلين من قتلى أحد	0	69	77	0	69
باعه إلا أن يشترط المبتاع	2	39	39	0	47
الرهن يركب بنفقته	0	67	58	2	82
لهم عذاب أليم	0	70	58	0	75
تدع الناس من الشر	6	9	9	0	12
من أعتق عبدا بين اثنين	1	49	31	9	56
ماله قيمة العدل	1	81	46	4	87
أعتق نصيبا، أو شقيصا	1	10	10	0	15
الأعمال بالنية، ولامرئ ما نوى	0	13	13	1	13
بيع الولاء وهبته	1	35	34	1	45
ركع ثم رفع رأسه	0	28	25	3	33
طاف بين الصفا والمروة	0	91	91	1	101
إذا كان أحدكم يصلي، فلا يبصق قبل وجهه	0	33	33	1	48
شرار الخلق عند الله يوم القيامة	6	46	26	2	62
أي العمل أحب إلى الله	1	65	60	0	94
شدة الحر من فيح جهنم	0	51	54	2	63
يفعل ذلك حين يكبر للركوع، ويفعل ذلك إذا رفع رأسه من الركوع	6	40	40	2	49
يقرأ في الظهر والعصر	6	46	42	0	57

حكم الالتفات في الصلاة	1	45	40	0	69
وما كان ربك نسيا	0	17	13	0	20
سورتين في كل ركعة.	8	19	18	7	46
فركع قبل أن يصل إلى الصف	8	66	66	0	74
لأشبهكم صلاة برسول الله	0	67	67	1	84
سبحانك اللهم ربنا وبحمدك	0	90	88	0	93
إذا نعس أحدكم وهو يصلي قليرقد	2	41	27	0	57
أيهم أكثر أخذا للقرآن	7	78	72	1	81
حج البيت، فلم يرفث	0	55	50	1	70
رأيتني دخلت الجنة	1	40	40	2	43
يعتكف في المسجد الحرام	2	100	91	5	100
لا يأكل إلا من عمل يده	0	33	32	1	35
الحلف بالله في السوق	6	8	10	0	13
تقاتلون قوما نعالهم الشعر	1	49	5	3	55
الطواف بالكعبة و الخوف من الله	0	18	12	0	23
لا تسافر المرأة ثلاثة أيام	0	17	17	1	20
العمل و حسن الخلق	1	15	13	1	20
قيام الليل و الثلث الاخير	1	15	12	2	17

[1] J. LEON, « Le traitement automatique des langues », CNRS, Université Paris 7, 2001 (France)

[2] F. YVON, « Une petite introduction au traitement Automatique du langage naturel », support de cours, Ecole Nationale Supérieur des télécommunications, Avril 2007

[3] P. BOUILLON, « Traitement automatique des langues naturelles », édition Duculot, 1998. (France)

[4] J.Y.ANTOINE, « Pour une ingénierie des langues plus linguistique », support de cours, 2005.

[5] L. TUERLINCKX, « La lemmatisation de l'arabe non classique », U.C.L. institut orientaliste (Belgique).

[6] Haifa Zargayouna, « Indexation sémantique de documents XML », Thèse de Doctorat en sciences à l'université Paris XI Orsay, 2005.

[7] P. Blanca, « La lemmatisation de l'arabe », GIRCSE (Groupe Interdisciplinaire de Recherche par Ordinateur sur les Signes d'Expression), université catholique de Milan, (Italie). .

[8] F. YVON, « Introduction au Traitement Automatique des Langues Naturelles », support de cours, 2006.

[9] Fouad Dahak, « Indexation des documents semi-structurés: Proposition d'une approche basée sur le fichier inversé et le Trie », Mémoire de Magister, Institut National de formation en Informatique (I.N.I), 2005.

[10] Jian-Yun Nie , « Le domaine de recherche d'information – Un survol d'une longue histoire», Département d'informatique et recherche opérationnelle Université de Montréal ;

[11] S. BALOUL, "Développement d'un système automatique de synthèse de la parole à partir du texte arabe standard voyellé", thèse de doctorat, université du Mans, 2003 (France).

[12] Y. HOCEINI, « Un système d'analyse morphologique de la langue arabe », mémoire magister, école nationale supérieure d'informatique, 2002 (Algérie).

[13]: د. ن علي،"اللغة العربية و الحاسوب" ، دار التعريب للنشر، 1988.

[14] Caroline Tabbellini « Un système de recherche d'information adapté aux données incertaines :adaptation du modèle de langue », Thèse de Doctorat à l'université Joseph Fourier-Grenoble I, 2007.

[15] Nawel Nassr,« Croisement de langues en recherche d'information : traduction et désambiguïsation de requêtes », Thèse de Doctorat à l'Université Paul Sabatier, 2002.

[16] Introduction a la RI ;

[17] Massih-Reza Aminni, « recherche et extraction d'information » ;

[18] Tahar DILEKH , « Implémentation d'un outil d'indexation et de recherche des textes en arabe », Mémoire Magister ;

[19] Siham Boulaknadel « Traitement Automatique des Langues et Recherche d'Information en langue arabe dans un domaine de spécialité :Apport des connaissances morphologiques et syntaxiques pour l'indexation », Thèse de doctorat , Université de Nantes,2008.

[20] Lobna HLAOUA, « Reformulation de Requêtes par Réinjection de Pertinence dans les Documents Semi-Structurés », Thèse de doctorat à l'Université Paul Sabatier, 2007.

[21] Ferradji Mohamed Abderraouf, « recherche d'information semi-srtucturée :modèle vectoriel », Université El-Hadj Lakhdar – BATNA,2011.

[22] M.Hammache Arezki, « Recherche d'information: un modèle de langue combinant mots simples et mots composé»,Thése deDOctorat en informatique, Université Mouloud Mammeri de Tizi-Ouzou.

[23] Herzallah Abdelkarim, , « recherche d'information»