كلية العلوم و التقنيات فاس
ⵜⴰⵖⵚⵍⵓⵍⵜ ⵏ ⵜⵇⵇⵓⵔⵉⵙ ⴰ ⵜⵓⵙⵙⵜⵙⵏ
Faculté des Sciences et Techniques de Fès

جامعة سيدي محمد بن عبد الله
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⴰⵎⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ
Université Sidi Mohamed Ben Abdellah

Centre d'Etudes Doctorales : Sciences et Techniques de l'Ingénieur

N° d'ordre 31/2018

THESE DE DOCTORAT

Présentée par

M. Issam SAHMOUDI

Discipline : Informatique

Spécialité : Fouille de textes

Sujet de la thèse : Contributions à l'Accès à l'Information en Langue Arabe : Regroupement Thématique des Résultats de Recherche et Indexation à base des Phrases-Clés.

Formation Doctorale : Sciences de l'ingénieur Sciences Physiques, Mathématiques et Informatique.

Thèse Présentée et soutenue le 07/07/2018, devant le jury composé de :

Nom Prénom	Titre	Etablissement	
Nour Eddine RAISS	PES	Faculté des Sciences Dhar El Mehraz de Fès	Président
Karim BOUZOUBAA	PES	Ecole Mohammedia des Ingénieurs de rabat	Rapporteur
Azzedine MAZROUI	PES	Faculté des Sciences d'Oujda	Rapporteur
Abderrahim BENABBOU	PH	Faculté des Sciences et techniques de Fès	Rapporteur
Ahmed ZINEDINE	PH	Faculté des Sciences Dhar El Mehraz de Fès	Examinateur
Abdelhak LEKHOUAJA	PES	Faculté des Sciences d'Oujda	Examinateur
Violetta CAVALLI-SFORZA	PH	Al Akhawayn University Ifran	Examinateur
Abdelmonaime LACHKAR	PES	Ecole Nationale des Sciences Appliquées de Tanger	Directeur de thèse

Laboratoire d'accueil : Laboratoire d'Ingénierie, Systèmes et Applications.

Etablissement : Ecole Nationale des Sciences Appliquées de Fès

بسم الله الرحمن الرحيم

﴿قال ربِّ إني وهن العظْم مني واشتعل الرأس شيبا ولم أكن بدعائك ربِّ شقيا﴾

صدق الله العظيم

﴿سورة مريم، الآية 3﴾

Dédicaces

Louange à ALLAH de m'avoir donné la puissance pour accomplir ce travail de recherche. C'est avec sa grâce que ledit travail a été accompli.

Je dédie cette thèse à :

La mémoire de mes Grands-Parents : Lahcen, Khdija, Abdessalam et Fatima, Le destin ne nous a pas laissé le temps pour jouir ce bonheur ensemble, Puisse ALLAH, le puissant les accueillir dans son saint paradis.

« A mes parents »,

Qui n'ont cessé d'investir pour que je puisse atteindre ce niveau,

M. Mohamed, mon cher père, l'homme le plus généreux du monde, qui n'a jamais tardé de m'encourager, et de me soutenir, ses aides et ses recommandations m'ont souvent incité à persévérer dans l'effort et à progresser dans ma vie universitaire et professionnelle.

Mme Aicha, ma chère mère, le plus grand cœur que j'ai rencontré, le rayon de soleil qui ne cesse de m'éclairer le chemin dans ma vie, sans son affection envers moi, ses oraisons pour moi, et surtout sans son sourire extraordinaire, je n'aurais pas pu frayer mon chemin et continuer à savourer le goût de la réussite et du succès.

« A ma chère femme»,

Mme Safae, qui m'a aidé et m'a soutenu moralement et sentimentalement pour que je puisse couronner mes recherches avec le présent rapport. Elle constitue ma ressource de joie et d'espoir, ses paroles me redonnent souvent confiance en mes aptitudes, et me comblent d'optimisme.

« A mes frères »,

M. Ahmed, M. Abdelilah et M. Amine, mes frères, qui n'ont cessé de m'encourager et de me soutenir durant toute cette période d'étude.

« A mes sœurs »,

Mme Hayte, Mme Soumia et Mme Jihane, mes adorables sœurs, leur précieux soutien m'a beaucoup servi à continuer jusqu'à ce point.

A Ma deuxième famille la famille ZERRIK : M. Mohamed, Mme Naima ; mes beaux-frères Bilal, Mouad, Soufiane, Omar et le petit Ismail.

A toute ma famille, A tous mes amis

Merci…

Issam SAHMOUDI

Remerciements

Le travail de cette thèse est le produit de plusieurs années de recherche durant lesquelles de nombreuses personnes ont marqué leurs empruntes clairement dans les traces réalisées pendant cette période et que je tiens à les remercier.

Je tiens tout d'abord à remercier mon directeur de thèse, Monsieur Abdelmonaime LACHKAR, professeur à l'Ecole Nationale des Sciences Appliquées de Tanger (Ex-Professeur à ENSA-USMBA, Fès), qui a bien voulu m'aider et m'encadrer tout au long de la réalisation de ce travail malgré ses tâches très lourdes. Je tiens à le remercier vivement pour ses conseils fructueux qu'il m'a prodigués, pour son orientation ciblée, pour sa disponibilité, même à des heures peu adaptées, et pour la confiance qu'il a bien voulu m'accorder.

Merci également aux membres de jury, pour avoir accepté de juger ce travail auquel ils ont accordé un intérêt particulier.

Je remercie toutes les personnes qui ont participé de manière directe ou indirecte à la concrétisation de ce travail. Qu'ils trouvent ici une expression de ma reconnaissance.

Publications de l'Auteur

Journaux :

[1] **Issam SAHMOUDI** and Abdelmonaime LACHKAR, "Formal Concept Analysis for Arabic Web Search Results Clustering," ***Journal of King Saud University - Computer and Information Sciences archive, Volume 29 Issue 2, April 2017, Pages 196-203***, ***10.1016/j.jksuci.2016.09.004***

[2] **Issam SAHMOUDI**, Hanane FROUD and Abdelmonaime LACHKAR, " A New Keyphrases Extraction Method Based On Suffix Tree Data Structure For Arabic Documents Clustering", ***International Journal of Database Management Systems (IJDMS)*** ,Vol.5, No.6, Decembre 2013.

[3] **Issam SAHMOUDI**, Abdelmonaime Lachkar," Clustering Web Search Results for Effective Arabic Language Browsing", ***International Journal on Natural Language Computing (IJNLC)*** Vol. 2, No.2, April 2013.

Communications :

[1] **Issam SAHMOUDI**, Abdelmonaime LACHKAR, "Modified KpST a new Arabic Keyphrases Extraction System ", ***LISA-WP'2016,*** 26 Avril 2016, Fès.

[2] **Issam Sahmoudi** and A. Lachkar, "Towards a linguistic patterns for arabic keyphrases extraction," ***2016 International Conference on Information Technology for Organizations Development (IT4OD),*** 2016.

[3] Mohammed BEKKALI**, Issam SAHMOUDI** and Abdelmonaime LACHKAR,"Enriching Arabic Tweets Representation based on Web Search Engine and the Rough Set Theory", ***The 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining(ASONAM 2015),*** 25-28 Aout 2015, Paris.

[4] ***Issam SAHMOUDI***, Abdelmonaime LACHKAR, "IMPROVED KPST SYSTEM FOR ARABIC KEYPHRASES EXTRACTION", ***Journées Doctorales sur l'Ingénierie de la Langue Arabe (JDILA'2015),*** 28-29 Octobre 2015, Fès.

[5] **Issam SAHMOUDI**, Abdelmonaime LACHKAR, " Performance Evaluation of Text Based Keyphrases Representation for Arabic Text Mining Appilications", ***The 5th Intetrnational Conference on Arabic Language Processing (CITALA'14),***26-27 November 2014, CITALA, Ouajda.

[6] **Issam SAHMOUDI**, Abdelhamid Lachkar, Said ALAOUI OUATIK, Abdelmonaime LACHKAR, “Improving Arabic Text Categorization Based On Keyphrases Extraction And Association Rules Mining”, ***Les Journées Doctorales en Technologies de l'Information et de la Communication (JDTIC’14)***,19-20 Juin 2014, ENSIAS, Rabat.

[7] **Issam SAHMOUDI**, Abdelmonaime LACHKAR, “Interactive System Based on Web Search Results Clustering for Arabic Query Reformulation”, ***IEEE international Colloquium in Information Science and Technology (IEEE CIST’14)***, 20-22 Octobre 2014,Tetouan-Chefchaouen.

[8] **Issam SAHMOUDI**, Hanane FROUD and Abdelmonaime LACHKAR,” Performance Evaluation Of Our Proposed Keyphrases Extraction Method Kpst And The Kp-Miner System For Arabic Language ”, ***The Fifth Workshop On Information Technologies And Communication (WOTIC 2013)***,26-27 Decembre 2013, Fès.

[9] Hanane FROUD, **Issam SAHMOUDI** and Abdelmonaime LACHKAR, “An Efficient Approach to Improve Arabic Documents Clustering Based on a new Keyphrases Extraction Algorithm”, ***Second International Conference of Data Mining and Knowledge Management Process(CDKP***), 2-3 November 2013,, Dubai, UAE.

الملخص

هذه الأطروحة هي جزء من مشروع يهدف إلى إنشاء وتحسين مكونات فهرسة نظم البحث عن المعلومات للغة العربية من أجل تفادي المشاكل التي تعرفها هذه اللغة في مجال تنقيب النص عامة. من خلال دراسة النظم القائمة كنا قادرين على تحديد وتصنيف كل هذه المشاكل إلى صنفين مشاكل تتعلق بمشكلة التصفح وأخرى بمشكلة الفهرسة .

فبالنسبة لمشكلة التصفح، الأنظمة الحالية مثل جوجل «*Google*» ، ياهو «*Yahoo*» و بنج «*Bing*» تقوم بارجاع قائمة من عشرات الآلاف من « *Snippet* » لمستخدمي الويب، حيث أغلبهم يقومون بتصفح فقط الصفحات الأولى، وبالتالي فإن الوثائق في نهاية القائمة لا يتم تصفحها أبدا على الرغم من أنها قد تكون ذات صلة و أهمية للمتصفح.

أما مشكلة الفهرسة، فالفهرسة على الكلمات الأساسية تعاني من مشكلة الغموض، الأمر الذي يؤثر سلبا على نتائج نظم البحث عن المعلومة للغات مختلفة عموما واللغة العربية على وجه الخصوص.

المساهمات المقدمة لدعم مجال البحث عن المعلومة في اللغة العربية وكجزء من هذه الأطروحة قمنا باقتراح نظام تفاعلي يعتمد على خوارزمية *STC*، يعمل على تجميع نتائج محركات البحث من أجل تصفح أسهل وأحسن. وكمساهمة ثانية قمنا باقتراح نظام يستند على *FCA* يعمل على تجميع نتائج محركات البحث ويوفر واجهة تصفح هرمي على مستويين. ولحل مشكل الفهرسة اقترحنا نظام يسمى *KpST* يستند على خوارزمية *STC* يعمل على استخراج الجمل الأساسية. بعد ذلك قمنا بتحسينه وذلك بإضافة قواعد لغوية، واستخدام *C-Value* .

Résumé

Cette thèse s'inscrit dans le cadre d'un projet, qui vise à améliorer les différents composants d'un *Système d'Indexation et de Recherche d'Information pour la langue arabe*, dans le but de remédier aux différents problèmes résultant de la complexité de cette langue dans le domaine de *la fouille de textes*. Pour cette raison, dans notre travail et lors de l'élaboration de l'état d'art et l'étude des systèmes existants, nous avons pu recenser et catégoriser l'ensemble des problèmes liés d'une part au processus de consultation des résultats de recherche web et d'autre part au processus d'indexation des documents. Notons que dans le cadre de la consultation, les moteurs de recherche existants tel que Google, Yahoo, Bing retournent une liste ordonnée d'une dizaine de milliers de snippets (métas-données), les utilisateurs ne consultent que les premières pages, et par conséquent les documents situés à la fin de la liste que très rarement consultables bien qu'ils puissent être pertinents. Au niveau du processus d'indexation, la méthode d'indexation basée sur les mots-clés pose un problème d'ambiguïté, ce qui influence négativement les résultats des systèmes de recherche d'information pour les différentes langues en particulier la langue arabe.

Pour remédier aux problèmes de consultation et d'indexation nous proposons dans le cadre de cette thèse différentes contributions pour soutenir le domaine de *Recherche d'Information* pour la langue arabe. Nous avons commencé par un système basé sur l'algorithme *STC « **Suffix Tree Clustering** »*, permettant le regroupement thématique des résultats de recherche pour les utilisateurs arabes, une deuxième contribution est un système basé sur le *FCA* « ***Formal Concept Analysis*** » qui permet un regroupement conceptuel et fournit une interface de consultation hiérarchique sur deux niveaux. Après, nous avons proposé une nouvelle approche qui permet l'extraction de ***phrases-clés***, basé sur l'algorithme ***d'arbre de suffixes*** dans un nouveau système nommé *KpST*. Par la suite, nous avons apporté des améliorations au système *KpST*, en ajoutant une couche de filtrage linguistique, et en utilisant une nouvelle mesure pour le calcul de score basée sur la *C-Value*, ce système est nommé *improved-KpST*.

Abstract

This thesis is part of a project aimed at creating and improving the various components of an *Arabic Information Retrieval System* in order to resolve the various problems due to the complexity of this language in the field of Text Mining applications. For this reason, after the study of the existing systems, we were able to identify and categorize all the problems linked on the one hand to the process of consultation and on the other hand to the indexing process.

For the **Browsing problem,** existing search engines such as Google, Yahoo, Bing return an ordered list of thousands of snippets users only consult the first pages, and consequently the documents located at end of the list will never be consulted despite that they may be relevant.

As well as the **problem of indexing,** keywords indexing poses a problem of ambiguity, which negatively affects the results of information retrieval systems for different languages especially for the Arabic language.

The various contributions were proposed in this thesis project to support the field of *Arabic Information Retrieval System*. We began with an interactive system based on the *Suffix Tree Clustering* (*STC)* algorithm to cluster the web search results for Arabic users. A second contribution is a system based on the *Formal Concept Analysis* (*FCA*), which allows conceptual clustering and provides a Hierarchical consultation interface on two levels. Next, we proposed a new approach based on the suffix tree algorithm that allows the extraction of keyphrases named *KpST*, afterwards we made improvements to the KpST system by adding a layer of linguistic filter, and using a new metric based on *C-Value* for score calculation, and this system is named *Improved-KpST*.

Table des matières

Liste des tableaux

Liste des figures

Introduction générale

1 Contexte Général

La recherche d'information (RI) n'est pas un domaine récent, il date des années 40. Une des premières définitions de la RI a été donnée par Salton [1]: « *la recherche d'information est un domaine qui a pour objectif, la représentation, l'analyse, l'organisation, le stockage et l'accès à l'information* ». Plusieurs tâches se regroupent sous le vocable de la RI, la plus ancienne est la recherche documentaire, nous y trouvons également d'autres tâches plus au moins récentes comme : le filtrage de l'information, l'extraction de l'information, la recherche d'information multilingue, les systèmes des questions réponses, la recherche d'information sur le web, etc.

Les systèmes de recherche d'information (SRI) dédiés aux documents web de la langue arabe ont connu une croissance similaire aux SRI des autres langues. Cependant, notre étude des systèmes existants pour l'indexation et la recherche de l'information à travers le web arabe a pu dégager plusieurs problématiques auxquels nous nous intéressons dans le cadre de ce travail. La majorité de ces inconvénients sont dus aux particularités de la langue arabe. En effet, l'une des particularités principales engendrant une partie de ces problèmes est la non diacritisation des textes car les mots de la quasi-totalité des textes actuels sont non diacritisés (non voyellés). Or, sans diacritiques un mot en arabe devient ambigu et peut porter plusieurs sens et designer des concepts différents. Par exemple, le mot noir "**أسود**" et le mot lions "**أسود**" s'écrivent de la même façon en langue arabe. Les inconvénients recensés ont été alors catégorisés de la façon suivante :

Le problème de Consultation : les SRI existants tel que Google, Yahoo, Bing retournent une liste ordonnée de snippets. Tel qu'il a été reporté par [29], il est connu que les utilisateurs ne consultent que les premières pages, et par conséquent les documents situés à la fin de la liste ne sont que très rarement consultés, bien qu'ils puissent être pertinents. Or, à cause de non diacritisation des textes arabes, cette situation devient plus grave dans les cas où le sens voulu par l'utilisateur se retrouve vers la fin de la liste. Ainsi, la consultation sous forme de snippet n'est pas du tout convenable ni adéquate pour le cas de la langue arabe. Pour aborder ce problème, nous avons commencé par un état d'art sur les approches existantes pour les autres langues. La majorité des travaux de recherche étudiés proposent de regrouper les résultats de recherche sous thématiques en utilisant différentes méthodes. Alors, nous proposons d'étudier la possibilité d'adapter ces méthodes aux particularités de la langue arabe.

Le problème d'indexation : l'étape d'indexation est primordiale dans les processus des systèmes de recherche d'information (SRI). Généralement, les documents et les requêtes sont représentés comme des listes de mots clés pondérés dans un tel système. L'appariement document-requête est lexical et se base sur la présence ou l'absence d'un mot de la requête dans le document. Or, un même mot peut désigner différents concepts (et donc exprimer différents sens) et différents mots

peuvent avoir une même signification. De ce fait, L'appariement lexical ne considère pas ces aspects. Conséquemment, un document pertinent, peut contenir des synonymes des mots de la requête, et par conséquent les documents ne sont pas retrouvés. En contrepartie, des documents non pertinents, contenant des mots lexicalement identiques mais sémantiquement différents (homonymes) des termes de la requête, sont retournés à l'utilisateur. Pour pallier à la résolution de cette problématique, nous avons commencé par un état d'art des travaux existants. Différentes approches ont été proposé pour améliorer l'indexation à base des mots clés. Les ***phrases-clés*** « *KeyPhrase* », présentent une grande utilité dans les domaines de fouille de textes « *Text-Mining* » et d'extraction des connaissances, permettent de nettoyer le texte et d'extraire l'information essentielle. Dans le cadre de cette thèse, nous allons étudier comment intégrer les phrases-clés dans l'étape d'indexation afin de contribuer à la solution de la problématique citée plus-haut.

2 Objectif

Cette thèse s'inscrit dans le cadre d'un projet qui vise à créer et améliorer les différentes composantes d'un Système d'Indexation et Recherche d'Information pour la langue arabe, dans le but de remédier aux différents problèmes menés par la complexité de cette langue dans le domaine de de fouille de textes.

3 Organisation de la thèse

3.1 Présentation générale

Le présent rapport est composé de trois parties comme illustré sur la figure 1, précédées par une introduction générale, et suivies d'une conclusion et des perspectives. La première partie intitulée *Etat de l'Art*, composée des 2 premiers chapitres, traite de l'impact du traitement automatique de la langue arabe en Système de Recherche d'information (SRI). La deuxième partie dénommée *Problème de Consultation* contient les chapitres 3 et 4. Elle regroupe les différentes contributions que nous avons proposées pour le regroupement thématique des résultats de recherche retournés par un SRI « ***Web search results clustering*** (WSRC) ». La troisième partie concerne le *Problème d'Indexation*. Elle est composée des derniers chapitres 5 et 6, et est consacrée à l'ensemble des contributions proposées pour remédier au problème d'indexation par des mots-clés.

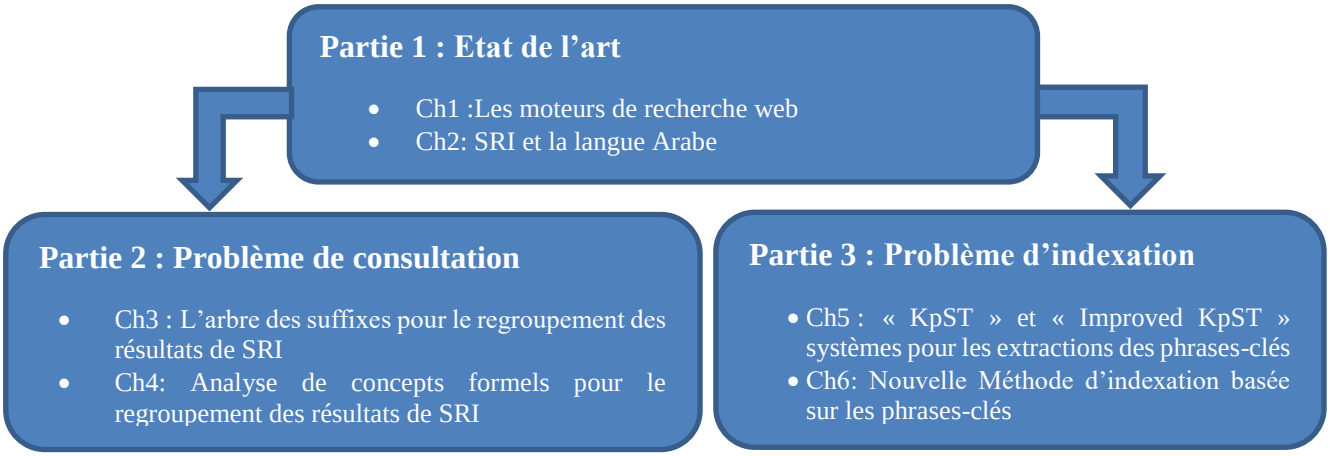


Figure 1 : *Structure de la thèse*

3.2 Contenu des chapitres

Chapitre 1 : Les moteurs de recherche web

Dans ce chapitre, nous introduisons les moteurs de recherche web. D'abord nous commençons par un aperçu général sur ces systèmes. Par la suite nous présentons l'architecture de ces systèmes, et vers la fin nous citons des exemples de ces derniers.

Chapitre 2 : SRI et la langue arabe

Dans ce chapitre, nous présentons les différents défis et problèmes liés aux SRI et à la langue arabe. Nous nous focalisons sur l'insuffisance des outils de recherche actuels, qui sont souvent mal adaptés aux spécificités de la langue arabe.

Chapitre 3 : L'arbre des suffixes pour le regroupement des résultats de SRI.

Dans ce chapitre, l'utilisation de l'algorithme d'arbre des suffixes pour le regroupement thématique des résultats de SRI est mise en évidence. Nous allons présenter notre modèle que nous avons proposé pour regrouper les résultats de SRI appliqué à la langue arabe, des évaluations sont présentées pour montrer la performance de la solution proposée.

Chapitre 4 : Analyse de concepts formels pour le regroupement des résultats de SRI.

Dans ce chapitre, nous présentons un nouveau modèle pour le regroupement thématique des résultats de SRI basé sur l'Analyse de Concepts Formels (Formal Concept Analysis : FCA) appliqué à la langue arabe. Nous commençons d'abord par le contexte mathématique de cette théorie, par la suite nous présentons notre modèle, enfin nous présentons les résultats d'évaluation de notre système proposé basé sur FCA.

Chapitre 5 : « KpST » et « Improved KpST » système pour les extractions des phrases-clés pour la langue arabe.

Dans ce chapitre, nous présentons notre modèle d'extraction des phrases-clés basé sur les arbres de suffixes appliqué à la langue arabe, ainsi que les améliorations que nous avons apportées au niveau

de la deuxième version de ce modèle. Vers la fin, nous montrons les résultats des différentes expériences qui ont été effectuées.

Chapitre 6 : Nouvelle méthode d'indexation basée sur les phrases-clés.

Dans ce chapitre, nous commençons par présenter les travaux connexes sur le processus d'indexation dans les systèmes d'indexation et recherche d'information (SIRI). L'objectif est de citer l'ensemble des problèmes liés à l'utilisation d'indexation basée sur les mots clés simples pour les langues latine d'une manière générale et en particulier la langue arabe. Ainsi, Nous présentons les différentes solutions dans la littérature pour surmonter ces problèmes. Par la suite, nous présentons notre SIRI dédié à la langue arabe. Ce dernier est basé principalement sur une nouvelle méthode d'indexation utilisant les phrases-clés au lieu des mots clés pour améliorer les performances des SRI.

Partie 1 : Etat de l'art

- **Chapitre 1 : Les moteurs de recherche web**
- **Chapitre 2 : SRI et la langue Arabe**

Introduction de la partie1

Dans cette partie, nous présentons les problèmes liés aux moteurs de recherche. Le premier problème concerne la consultation des résultats de recherche, le deuxième problème concerne l'indexation. Elle comprend deux chapitres, le premier est un état de l'art sur les moteurs de recherche web, où nous commençons par un aperçu général sur ces systèmes et leurs objectifs. Par la suite, nous présentons l'architecture de ces systèmes. Vers la fin nous citons des exemples de ces derniers. Le deuxième chapitre présente l'ensemble des problèmes liés aux systèmes de recherche d'information (SRI) avec la langue arabe. Nous listons ces problèmes en suivant une étude analytique et qualitative de la RI en langue arabe, en mettant l'accent sur l'insuffisance des outils de recherche actuels, qui sont souvent mal adaptés aux spécificités de la langue arabe.

Finalement, cette partie se termine par une conclusion qui met l'accent sur les différents points traités dans les deux chapitres.

CHAPITRE 1. Les moteurs de recherche web

1 Introduction

Les moteurs de recherche web sont des outils primordiaux pour récupérer des informations à partir du Web dans le cadre d'un système de recherche d'information. La réponse à une requête de l'utilisateur retourne une liste de résultats classés par ordre de pertinence. L'utilisateur commence par consulter le début de la liste jusqu'à ce que l'information recherchée soit trouvée. Les moteurs de recherche web sont souvent efficaces pour certaines tâches de recherche telles que la recherche sur des organisations ou des établissements. Cependant, ils peuvent être moins efficaces pour satisfaire les requêtes en langue arabe. En effet, La recherche d'information en langue arabe montre souvent une certaine dissymétrie entre l'indexation et le traitement des requêtes provoquée particulièrement par l'absence des voyelles dans les textes arabes écrits et aussi par la nature agglutinante de l'écriture arabe.

Les résultats retournés sur différents thèmes ou sens de la requête sont mélangés dans la liste, alors l'utilisateur doit consulter un grand nombre d'éléments pour localiser ceux qui portent un intérêt pour lui. Ainsi, il n'y a aucun moyen pour déterminer exactement ce qui est pertinent pour l'utilisateur.

Par exemple, lors de l'indexation d'un document, le verbe "écrire" (**كَتَبَ**), le nom "livres" (**كُتُب**) et le nom "écrit" (**كَتْب**) sont tous indexés sous une seule et même entrée (**كتب**, ktb), car ils ne sont généralement pas vocalisés dans le texte. Il en est de même pour le mot **علم** qui peut désigner plusieurs sens (drapeau, science, connaître, etc) [29].

Une approche différente de la recherche d'information classique basée sur la pertinence des documents, la recherche d'information basée sur la catégorisation de la totalité des documents web, que ça soit manuellement ou automatiquement, permet à l'utilisateur de voir les résultats associés aux thématiques qui correspondent le mieux à sa requête. Cependant, même les plus grands répertoires web tels que l'Open Directory Project[1] ne couvrent qu'une petite fraction des pages existantes. De plus, le répertoire peut être utile pour une requête d'utilisateur particulier ou pour un aspect particulier de celui-ci. En effet, les répertoires web sont le plus souvent utilisés pour influencer la sortie d'une recherche directe en réponse aux requêtes des utilisateurs communs. Une troisième approche de la recherche d'information consiste à regrouper les résultats retournés par un moteur de recherche dans une hiérarchie de groupes étiquetés.

Dans ce chapitre, nous commençons par expliquer la structure interne des moteurs de recherche web. Ensuite, nous procédons à une étude des approches les plus communes à la consultation des résultats de la recherche sur le web. Nous terminons ce chapitre par un état d'art sur les différents systèmes de regroupement thématique des résultats de recherche.

[1] http://www.dmoz.org/

2 Structure interne des moteurs de recherche

Le moteur de recherche est un outil automatique pour collecter et indexer un grand nombre de pages web ; il recense des sites et pas des annuaires. Un moteur de recherche est foncièrement constitué de deux processus complémentaires (Figure 2) :

1. Un processus d'indexation du contenu du Web.
2. Et un processus de consultation de l'index ainsi constitué.

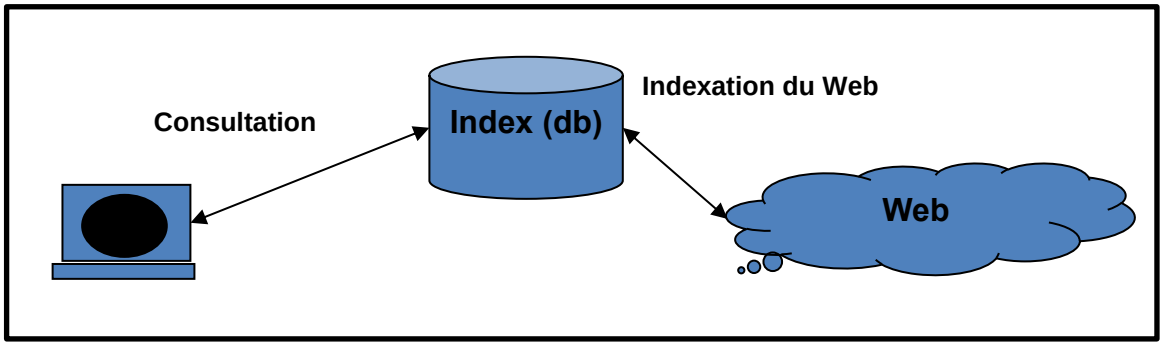


Figure 2 : *L'architecture fondamentale d'un moteur de recherche*

2.1 Définitions d'un système d'indexation et recherche d'information

Haddad [2] a présenté plusieurs définitions d'un *Système de Recherche d'Information (SRI),* qui désigne dans cette thèse un *Système d'Indexation et Recherche d'Information,* qui sont plus ou moins proches.

Tomek Strzalkowski [3] définit un *Système de Recherche d'Information* comme suit: « La tâche typique de la recherche d'information, est de sélectionner des documents dans une base de données, en réponse à une requête de l'usager, et leur rangement par ordre de pertinence». Tandis que *Alan Smeaton* donne la définition suivante [4] : « *Le but d'un Système de Recherche d'Information est de retrouver des documents en réponse à une requête des usagers, de manière à ce que les contenus des documents soient pertinents au besoin initial d'information de l'usager*».

Or, ces définitions restent insatisfaisantes [2] car elles n'explicitent pas les procédures de traitement de l'information ou de l'indexation automatique des documents. Ce sont des définitions qui prennent en compte ce que les utilisateurs perçoivent d'un *Système de Recherche d'Information* et ce qu'un *Système de Recherche d'Information* doit leur offrir. Or, il y a tout un ensemble de procédures pour que les usagers puissent accéder à l'information. *Salton* et *Mcgill* donnent une définition d'un *Système de Recherche d'Information* plus simple mais plus précise et complète [1] :

«*Un Système de Recherche d'Information traite de la représentation, du stockage, de l'organisation et de l'accès aux éléments de l'information* ».

Haddad [2] définit un *Système de Recherche d'Information*, illustré dans la figure 3, comme étant un système composé d'une part d'un module chargé du traitement, de l'indexation et du stockage de l'information. Où, le module indexation, construit à partir du traitement de l'information, est une structure de données organisées de manière à permettre l'accès rapide à l'information. D'autre part, il est composé d'un module qui sert à interagir avec les utilisateurs, doté des mécanismes de sélection d'information orientés par les requêtes des utilisateurs : le module interrogation. Enfin, un module de correspondance (fonction de correspondance) établit une association entre la requête de l'utilisateur et les documents traités.

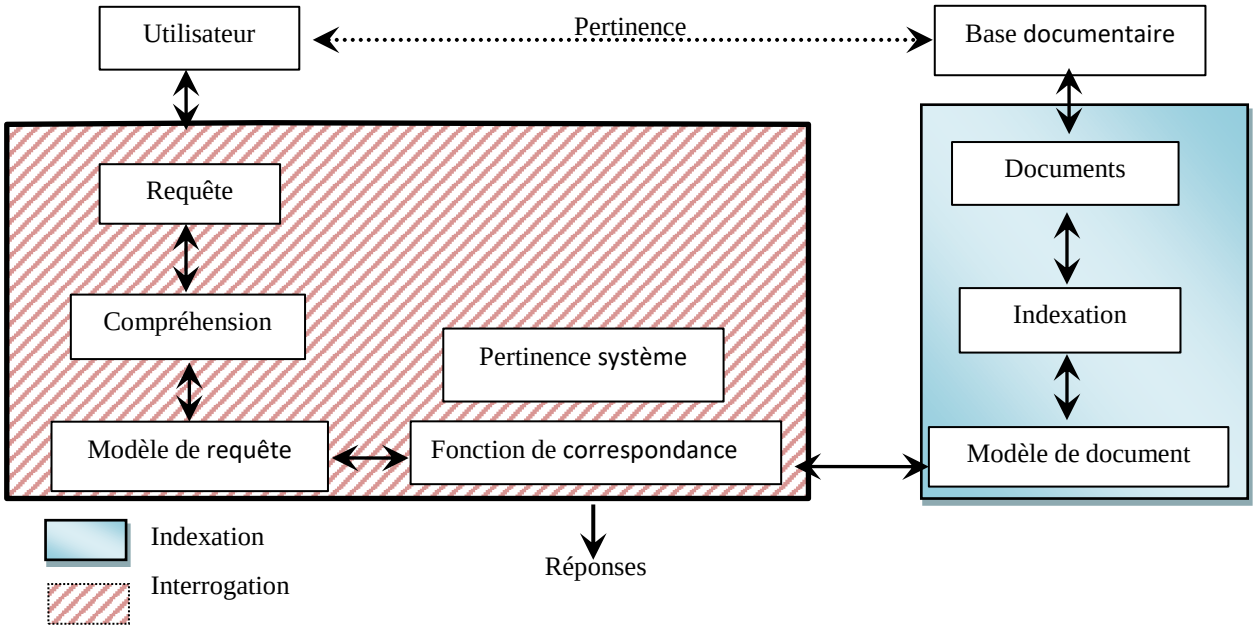


Figure 3 : *Architecture générale d'un système de Recherche d'Information* [2]

2.2 Modèles de recherche de documents textuels

La définition élémentaire de la recherche documentaire est l'ensemble des étapes permettant de chercher, identifier et trouver des documents relatifs à un sujet par l'élaboration d'une stratégie de recherche. Ces documents pourraient être de n'importe quel type de texte principalement non structurés, tels que des notices bibliographiques, des articles de journaux ou des paragraphes dans un manuel. Les requêtes des utilisateurs peuvent aller de descriptions complètes d'un besoin d'information à quelques mots. Cependant, cette définition n'est pas suffisamment informative, car un document peut être pertinent même s'il n'utilise pas les mêmes mots que ceux fournis dans la requête. L'utilisateur n'est généralement pas intéressé à récupérer des documents avec exactement les mêmes mots, mais avec les concepts que ces mots représentent [5].

La phase d'indexation des documents [6] est la première phase d'un *Système de Recherche d'Information*, et qui consiste à passer d'un document brut à une représentation qui sera utilisée lors de la recherche documentaire proprement dite, pour construire un index qui permettra de trouver facilement quel(s) document(s) contient(nent) quel(s) mot(s).

L'étape de la recherche documentaire peut alors commencer [6]. Lorsque les résultats d'une requête sont demandés, nous pouvons décomposer le processus en trois étapes : La requête est transformée dans une représentation interne au système de RI. Un score est calculé pour chaque document. Le *Système de Recherche d'Information* ordonne les documents en fonction de ce score et présente les *Np* premiers à l'utilisateur. Le score attribué à un document *d* pour une question *q* d'un modèle sera dans la suite noté *RSV(q; d)*. Le processus complet de la recherche documentaire est illustré sur l'exemple de figure 4 [6].

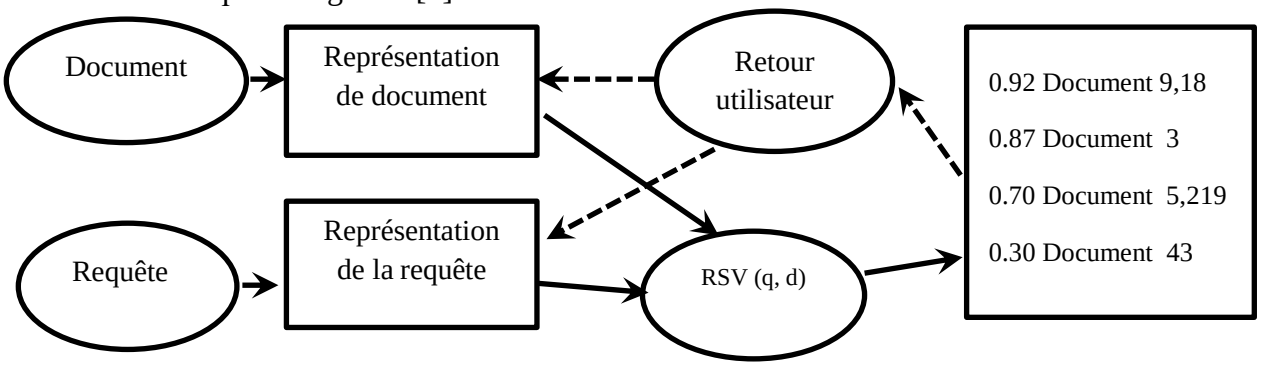


Figure 4 : *Processus de recherche documentaire* [6]

Piwowarski [6] distingue trois types d'approches en Recherche d'Information. L'approche ensembliste, encore très utilisée dans les systèmes opérationnels (les catalogues de bibliothèques), considère la recherche documentaire comme des opérations sur les ensembles de documents. L'approche algébrique envisage la recherche documentaire comme le calcul d'un score entre le document et la requête. Enfin, les approches probabilistes supposent que la pertinence d'un document est un phénomène incertain qui peut être représenté par un modèle.

Le rôle d'un modèle est d'abord de donner une signification au résultat de l'indexation. Un document est représenté par un ensemble de termes index et leurs poids. Si la plupart des chercheurs acceptent d'office qu'un terme index soit censé représenter un concept important décrit dans un document, il existe différentes façons d'interpréter son poids. Un modèle théorique doit donner une interprétation précise à ce poids. Le modèle doit aussi interpréter les relations possibles entre les termes d'indexation. Ces deux fonctions nous amènent à la représentation d'un document. Une représentation similaire peut être créée pour une requête.

Les modèles mathématiques [5] actuellement utilisés peuvent être classés en trois types: Booléen, statistique et linguistique. Un modèle est caractérisé par quatre paramètres :

- La représentation des documents et des requêtes ;
- La correspondance des stratégies pour évaluer la pertinence des documents à une requête de l'utilisateur ;
- Les méthodes de classement des réponses du modèle aux requêtes d'utilisateurs ;
- Les mécanismes de mesure de pertinence des résultats retournés par le modèle ;

Finalement, un modèle doit déterminer la relation entre un document et une requête à partir de leurs représentations, ceci se fait souvent avec un calcul de similarité.

2.2.1 Recherche booléenne

a) Modèle Booléen Standard

Dans cette section, nous présentons les principaux 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 Standard* [7].

Ce modèle est le plus simple de la recherche documentaire. Un document est représenté par l'ensemble des mots qu'il contient. Une requête est représentée comme une formule logique où chaque atome correspond à la présence d'un terme. Par exemple, *(((recherche ∨ extraction) ∧ information) ∧ (¬Arabe))* renverra les documents contenant le mot *« recherche »* ou le mot *« extraction »*, le mot *«information »* et qui ne contiennent pas le mot *«Arabe»*. Toutes les formules logiques sont valides dans ce formalisme.

	Le Modèle Booléen Standard
Objectif	• Capturer la structure conceptuelle et l'information contextuelle
Méthodes	• Coordination : AND, OR, NOT • Proximité • Racinisation (Stemming)/ Troncature
Avantages	• Facile à mettre en œuvre • L'expressivité et la clarté • Spécifications des synonymes (avec la clause OR) et des phrases (avec la clause AND)
Inconvénients	• Il est difficile de construire des requêtes booléennes • Tout ou rien ⇨AND trop sévère, et OR n'est pas suffisant • difficile à contrôler : sortie Null / surcharge • Pas de classement • Pas de pondération des termes d'index ou de la requête • Aucune mesure de l'incertitude

Tableau 1 : *Les caractéristiques essentielles, et les principaux avantages et inconvénients du Modèle Booléen Standard* [7].

Le tableau 1 récapitule les caractéristiques essentielles du *Modèle Booléen Standard* et la liste de ses principaux avantages et inconvénients [7]. Ce modèle dispose des atouts suivants :

1. Il est facile à mettre en œuvre et son calcul est efficace ;
2. Il permet aux utilisateurs d'exprimer des contraintes structurelles et conceptuelles pour décrire les importantes caractéristiques linguistiques. Les utilisateurs trouvent que les spécifications de synonymes (reflétée par la clause *OR*) et les phrases (représentés par des relations de proximité) sont utiles pour la formulation de requêtes ;
3. Il est très efficace si une requête nécessite une sélection exhaustive et sans ambiguïté ;

4. Il offre une multitude de techniques pour élargir ou affiner la recherche ;
5. Le *Modèle Booléen Standard* peut être particulièrement efficace dans les étapes ultérieures du processus de recherche, en raison de la clarté et la rigueur avec laquelle les relations entre les concepts peuvent être représentées.

Les principaux défauts de ce modèle d'après A. Spoerri et al. sont les suivants [7] : Les utilisateurs trouvent qu'il est difficile de construire des requêtes booléennes efficaces pour plusieurs raisons. Les utilisateurs utilisent les termes en langage naturel *AND, OR* ou *NOT* qui ont une signification différente lorsqu'ils sont utilisés dans une requête. Ainsi, les utilisateurs pourront faire des erreurs quand ils forment une requête booléenne. Pour illustrer ces défauts, considérons l'expression suivante *«A et B»*. Dans nos conversations ordinaires cette expression se réfère généralement à plus d'entités repérées par l'expression *«A»*, alors que lorsqu'elle est utilisée dans le cadre de la Recherche d'Information, elle se réfère à moins de documents que ce qui serait récupéré par *«A»*. Par conséquent, l'une des erreurs les plus courantes faites par les utilisateurs est de substituer l'opérateur logique *AND* par l'opérateur logique *OR* lors de la traduction d'une phrase en anglais à une requête booléenne. En outre, pour former des requêtes complexes, les utilisateurs doivent se familiariser avec les règles de préséance et l'utilisation des parenthèses. Enfin, les utilisateurs sont débordés par la multitude de façons avec lesquelles une requête peut être structurée ou modifiée, en raison de ces différentes combinaisons de requêtes le nombre de concepts augmente. En particulier, les utilisateurs ont du mal à identifier et appliquer les différentes stratégies qui sont disponibles pour réduire ou élargir une requête booléenne [7], [8].

1. Seuls les documents qui satisfont la requête exactement sont récupérés. D'une part, l'opérateur *AND* est trop sévère, car il ne fait pas de distinction entre le cas où aucun des concepts n'est satisfait et le cas où tous sauf un sont satisfaits. Par conséquent, aucun ou très peu de documents sont récupérés lorsque plus de trois ou quatre critères sont combinés avec l'opérateur booléen *AND* (dénommé le problème de *Null-Output*). D'autre part, l'opérateur OR ne reflète pas combien de concepts ont été satisfaits. Ainsi, souvent trop de documents sont récupérés (le problème de *Output Overload*).
2. Il est difficile de contrôler le nombre de documents trouvés. Les utilisateurs sont souvent confrontés à l'hypothèse du *Null-Output* ou au problème d'Output *Overload* et ils sont perdus pour le choix de la façon de modifier la requête pour récupérer un nombre de documents raisonnable.
3. Le *Modèle Booléen Standard* ne fournit pas un classement de la pertinence des documents trouvés.
4. Il ne représente pas le degré d'incertitude ou l'erreur due au problème de vocabulaire [8].

b) Techniques de rétrécissement et d'élargissement (Narrowing and Broadening Techniques)

Une requête booléenne peut être décrite en termes des quatre opérations suivantes [7] : le degré et le type de coordination, les contraintes de proximité, les spécifications sur le domaine de recherche et le degré de Racinisation (Stemming) des termes utilisés. Si les utilisateurs des systèmes basés sur le modèle booléen souhaitent reformuler une requête booléenne, alors ils ont besoin de faire des choix en se basant sur les opérations citées ci-dessus pour créer une requête qui est suffisamment large ou étroite en fonction de leurs besoins d'information. Toute requête peut être reformulée pour obtenir la précision et le rappel désirés. Dans la reformulation de requête chaqu'une des quatre opérations a son principal opérateur, dont certains ont tendance à avoir un effet rétrécissant ou élargissant. Pour chaque opérateur avec un effet de rétrécissement, il y a un ou plusieurs opérateurs inverses avec un effet d'élargissement [7].

c) Modèle Smart Boolean

Dans ce qui suit, le Modèle «Smart Boolean» développé par Spoerri [7] et *Marcus* [9], tente d'aider les utilisateurs à construire et modifier une requête booléenne en se basant sur les quatre dimensions qui caractérisent une requête booléenne. Le tableau 2 présente un résumé de ses principales caractéristiques.

	Le Modèle Smart Boolean
Objectif	• Recherche de Structure processus de (re-) formulation • Utilisez les bases de connaissances structurelles et contextuelles et la clarté des expressions booléennes
Méthodes	• L'énoncé en langage naturel est automatiquement traduit en représentation booléenne du sujet • Représentation des sujets booléens : ANDs ou ORs des concepts Mot-clés /Stem, tous les champs ➢ Info. Conceptuelle → Coordination et facteur d'ajout/Suppression ➢ Info. Contextuelle → Proximité ➢ Info. Structurelle → Niveaux de champ ➢ Synonyme ou word relationship → Racinisation « *Stemming* »/ Chevauchement de troncature ⇨ toute cette information peut être utilisée pour classer les documents • Techniques d'élargir et réduire d'une requête
Avantages	• On n'a pas besoin des opérateurs booléens • Aider l'utilisateur pour (re) formulation la requête : en posant des questions ciblées aux utilisateurs pour modifier automatiquement la requête • «Pourquoi ce n'est pas pertinent?» → active les méthodes de rétrécissement • «Élargir par la suppression des facteurs» pour estimer le rappel
Inconvénients	• Comment visualiser ? ➢ Représentation de la requête conceptuelle ➢ Techniques de modification de requêtes et leurs effets ➢ Retour de pertinence structuré

Tableau 2 *: Les caractéristiques essentielles et les principaux avantages et inconvénients du Modèle Smart Boolean* [7]

Dans ce qui suit, le Modèle «Smart Boolean» développé par Spoerri [7] et *Marcus* [9], tente d'aider les utilisateurs à construire et modifier une requête booléenne en se basant sur les quatre dimensions qui caractérisent une requête booléenne. Le tableau 2 présente un résumé de ses principales caractéristiques.

L'un des objectifs du Modèle «Smart Boolean» [7] est de faire usage des connaissances structurelles contenues dans les requêtes des utilisateurs, où les différents domaines présentés par ces requêtes représentent des contextes d'informations utiles. En outre, le Modèle «Smart Boolean» utilise le fait que plusieurs concepts peuvent partager une racine commune. Par exemple, les concepts «computing» et «computers» ont la même racine «comput» [7].

La stratégie initiale du Modèle «Smart Boolean» est de commencer avec la requête la plus large possible dans les limites de la façon dont les facteurs et leurs synonymes ont été coordonnées. Par la suite, il modifie la représentation de la requête en utilisant uniquement les racines des concepts. Une fois que la requête principale a été effectuée, les utilisateurs sont guidés dans le processus d'évaluation des documents récupérés [7]. Ils choisissent parmi une liste de raisons pour expliquer pourquoi ils estiment que certains documents sont pertinents. De même, ils peuvent indiquer pourquoi les autres documents ne sont pas pertinents en interagissant avec une autre liste de raisons possibles. Ce feedback d'utilisateur est utilisé par le Modèle «Smart Boolean» pour modifier automatiquement la représentation booléenne de la requête principale, ce qui est plus approprié. Le Modèle «Smart Boolean» offre un ensemble de stratégies pour modifier une requête basée sur le feedback reçu ou le besoin exprimé pour restreindre ou élargir la requête.

d) Modèles Booléens Etendus

Plusieurs méthodes ont été développées pour étendre le modèle booléen (tableau 3) pour répondre aux défis suivants :

1. Les opérateurs booléens sont trop stricts et des moyens doivent être trouvés pour les adoucir.
2. Le *Modèle «Smart Boolean»* ne contient aucune règle pour le classement [7].
3. Le *Modèle Booléen* ne supporte pas l'attribution de poids aux termes de la requête ou de document.

Le P-Norm et les approches de logique floue qui étendent le Modèle Booléen pour répondre aux défis ci-dessus seront brièvement décrits. La méthode P-Norm développée par Fox et al. 1983 [10] permet aux termes des requêtes et des documents d'avoir un poids. Ces poids normalisés peuvent être utilisés pour classer les documents dans l'ordre décroissant de distance entre le point (0, 0, .., 0) pour une requête OR, et dans l'ordre croissant de la distance à partir du point (1, 1,.., 1) pour une requête AND. En outre, les opérateurs booléens ont un coefficient associé P pour indiquer le degré de précision

de l'opérateur. Le P-Norm utilise une mesure basée sur la distance et le coefficient P détermine le degré d'exponentiation à être utilisé. L'exponentiation est un calcul coûteux, en particulier pour les valeurs P supérieures à un.

En théorie des ensembles flous, un élément a un certain degré d'appartenance à un ensemble au lieu du choix traditionnel d'adhésion binaire. Le poids d'un terme pour un document donné reflète le degré avec lequel ce terme décrit le contenu d'un document.

	Modèles Booléens Etendus
Objectif	• Les opérateurs booléens moins stricts • Sortie classée
Méthodes	• La logique floue [OR -> max], [AND -> min] and [NOT -> 1-max] (-) Le manque de sensibilité des valeurs min et max : min (0.2, 0.8) = min (0.2, 0.3)

Tableau 3 : *Les caractéristiques essentielles et les principaux avantages et inconvénients des Modèles Booléens Etendus* [7]

Par conséquent, ce poids reflète le degré d'appartenance du document dans l'ensemble flou associé au terme en question. Le degré d'appartenance à l'union (à l'intersection) de deux ensembles flous égale la valeur maximale (la valeur minimale) des degrés d'appartenance des éléments des deux ensembles. Dans le *Modèle «Mixed Min and Max»* développé par *Fox* et *Sharat* [11], la similarité requête document est une combinaison linéaire des min et max des poids des documents.

2.2.2 Modèle Statistique

Les modèles probabilistes et d'espace vectoriel sont les deux exemples majeurs de l'approche statistique de la Recherche d'information. Les deux modèles utilisent l'information statistique sous forme de fréquences des termes afin de déterminer la pertinence des documents par rapport à une requête. Le tableau 4 présente un résumé des principales caractéristiques des modèles cités ci-dessous. D'autres approches basées sur le modèle statistique seront discutées dans cette section.

a) Modèle de l'espace vectoriel (Vector Space Model « VSM »)

Le *Modèle d'Espace Vectoriel* représente les documents et les requêtes en tant que vecteurs dans un espace multidimensionnel, dont les dimensions sont les termes utilisés pour construire des indexes pour représenter les documents [1]. La création d'un index implique l'utilisation de l'analyse lexicale pour identifier les termes importants, où l'analyse morphologique qui permet de réduire les différentes formes de mots en une «racine» commune, et les occurrences de ces racines sont calculées. Les requêtes et les documents sont comparés en utilisant leurs vecteurs, en utilisant, par exemple, la mesure de similarité Cosinus. Dans ce modèle, les termes d'une requête peuvent être pondérés pour tenir compte de leur importance, et la pondération de chaque terme est calculée en utilisant sa distributions

statistiques dans la collection des documents et dans chaque document [1]. Le *Modèle d'Espace Vectoriel* peut attribuer un score important à un document qui ne contient que quelques termes de la requête, si ces termes se produisent rarement dans la collection des documents, mais fréquemment dans le document concerné.

b) Modèle probabiliste

Le *Modèle Probabiliste* est basé sur *«Probability Ranking Principle»*, qui suppose qu'un système de Recherche d'Information est censé classer les documents en fonction de leurs probabilités de pertinence par rapport à la requête [7]. Le principe prend en compte le fait qu'il existe une incertitude dans la représentation de l'information nécessaire et des documents. Il peut y avoir une variété de sources de données qui sont utilisées par les méthodes probabilistes, et le plus commun est la distribution statistique des termes dans les documents pertinents et non pertinents.

Les approches probabilistes ont les avantages suivants :

1. Ils fournissent aux utilisateurs un classement de pertinence des documents trouvés. Par conséquent, ils permettent aux utilisateurs de contrôler la sortie en fixant un seuil de pertinence ou en spécifiant un certain nombre de documents à afficher.
2. Les requêtes peuvent être plus faciles à formuler, car les utilisateurs n'ont pas besoin d'apprendre un langage de requête et peuvent utiliser le langage naturel.
3. L'incertitude dans le choix des concepts de la requête peut être représentée.

Cependant, les approches probabilistes ont les défauts suivants :

1. Ils ont un pouvoir expressif limité. Par exemple, l'opération NOT ne peut pas être représentée car seuls les poids positifs sont utilisés. Il est prouvé que pour N entrées seulement 2^{N-1} requêtes booléennes possibles peuvent être générées par les approches statistiques qui utilisent des sommes linéaires pondérées pour classer les documents. Ce résultat découle de l'analyse des réseaux de seuil linéaire ou perceptrons booléens [7].
2. L'approche statistique n'a pas la structure convenable pour exprimer les caractéristiques linguistiques importantes telles que les phrases.
3. Le calcul des scores de pertinence peut être coûteux en calcul.
4. Une liste linéaire classée fournit aux utilisateurs une vue limitée de l'espace d'information et il ne suggère pas directement comment modifier une requête si on en a besoin [7].
5. Les requêtes doivent contenir un grand nombre de mots pour améliorer les performances de la recherche. Comme dans le cas du *Modèle Booléen*, les utilisateurs sont confrontés au problème d'avoir à choisir les mots appropriés qui sont également utilisés dans les documents pertinents.

Modèle Statistique	Modèle de l'espace vectoriel	Modèle probabiliste
Motivation	Simplifier la formulation requête Possibilité de contrôler la sortie	S'adresser à l'incertitude dans les représentations de la requête
Objectif	Classez la sortie en se basant sur Similarité	Probabilité de la pertinence
Méthodes	Mesure de Cosine	L'utilisation de différents modèles
Source	**Statistiques des termes de la requête** Espace vectoriel : • Similarité $(Q,D)=\sum\left(W_{iq}\times W_{ij}\right)$ / "normalisateur" Tel que $W_{iq}=\left(0,5+0,5freq_{iq}/\max freq_{q}\right)\times idf\ (i)$ $W_{ij}=freq_{ij}\times idf\ (i)$ • Fréqu. inverse du terme dans la collection $idf\ (i)=\log_2\left(N-n(i)\right)/n(i)$. Probabiliste : • Poids du terme $=\log\left[\left(r_t/R-r_t\right)/\left(\left(n_t-r_t\right)/\left(\left(N-n_t\right)-\left(R-r_t\right)\right)\right)\right]$ = "(hits / misses) / (false alarmes / correct misses)" • Similarité $jk=\sum\left(C+idf\ (i)\right)\times tf\ (i,j)$ Tel que $tf\ (i,j)=K+\left(1-K\right)\left(freq(i,j)/\max freq(j)\right)$	
Issues	• Comment exprimer NOT ? Recherche de proximité ? • Puissance expressive limitée • Calcul intensif • Suppose que les termes sont indépendants Manque de structure pour représenter les entités linguistiques importantes Comment faire pour mieux visualiser l'ensemble récupéré ?	• Estimation des probabilités nécessaires • Une connaissance préalable nécessaire • hypothèse de l'indépendance • Relations booléennes perdues • Quel est le meilleur modèle ?

Tableau 4 : *Caractéristiques essentielles des Modèles d'Espace Vectoriel et Probabiliste* [7]

c) Modèle d'Indexation Sémantique Latente (Latent Semantic Indexing Model)

Plusieurs techniques statistiques et celles du domaine de la Recherche d'Information ont été utilisées en association avec l'information sémantique pour étendre le *Modèle d'Espace Vectoriel* pour aider à surmonter certains problèmes décrits ci-dessus, tels que le «problème de la dépendance» ou le «problème de vocabulaire». Un tel procédé est le *Modèle d'Indexation Sémantique Latente (LSI)*.

En LSI les associations entre les termes et les documents sont calculées et exploitées dans le processus de Recherche d'Information. L'hypothèse est qu'il y a une certaine structure «latente» dans l'usage des mots à travers les documents et les techniques statistiques qui peuvent être utilisées pour estimer cette structure latente. Un avantage de cette approche est que les requêtes peuvent récupérer des documents, même s'ils n'ont pas de mots en commun.

La technique *LSI* peut capturer la plus profonde structure associative que des simples corrélations terme à terme. La seule différence entre *LSI* et les autres *Modèles Vectoriels* est que *LSI* représente des

termes et des documents dans un espace de dimension réduite des dimensions d'indexation dérivées. Comme avec le modèle d'espace vectoriel, la pondération du terme et de la pertinence des feedback peut améliorer la performance *LSI* considérablement [7].

2.2.3 Approche linguistique

Dans la forme la plus simple de la recherche automatique du texte, les utilisateurs entrent des mots-clés qui sont utilisés pour rechercher les indexes inversés des mots-clés des documents. Cette approche récupère les documents en se basant uniquement sur la présence ou l'absence des mots indiqués par la représentation logique de la requête. Cette approche ne capture pas la signification complète ou profonde de la requête de l'utilisateur. Le Modèle *Smart Boolean* et les approches statistiques, chacun à sa manière spécifique, tentent de remédier à ce problème (voir le tableau 5). Les approches linguistiques ont également été développées pour résoudre ce problème en effectuant une analyse morphologique, syntaxique et sémantique pour récupérer des documents plus efficacement [8]. Dans une analyse morphologique, les racines et les affixes sont analysés pour déterminer la partie du discours (nom, verbe, adjectif, etc) des mots. Les phrases complètes doivent être analysées en utilisant une certaine forme d'analyse syntaxique. Enfin, les méthodes linguistiques doivent résoudre les ambiguïtés de mots et/ou de générer des synonymes pertinents ou quasi-synonymes basé sur les relations sémantiques entre les mots. Le développement d'un système de recherche linguistique sophistiqué est difficile et nécessite des bases de connaissances complexes de l'information sémantique et la recherche heuristique. Par conséquent, ces systèmes nécessitent souvent des techniques qui sont couramment désignés comme l'intelligence artificielle.

Niveau linguistique	**Recherche Booléenne**	**Statistique**	**Linguistique est fondée sur la connaissance**
Lexicale	Liste des Mots vides	Liste des Mots vides	Lexique
Morphologique	Symbole de troncature	Racinisation	L'analyse morphologique
Syntaxique	Opérateurs de proximité	Phrases	Phrases grammaticales
Sémantique	Thésaurus	Clusters des mots co-occurents	Réseau de mots/phrases dans relation-ships sémantiques

Tableau 5 *: Principales méthodes de recherche en termes de comment traiter les questions lexicales, morphologiques, syntaxiques et sémantiques* [7]

2.3 Consultation des résultats de recherche

La consultation des résultats de recherche consiste à afficher les résultats sur une interface web d'une manière qui aide les utilisateurs à identifier les documents cherchés. Ainsi, une interface web idéale ne doit pas orienter les utilisateurs par hasard vers les documents qu'ils jugeraient comme non pertinents. La tâche est particulièrement difficile du fait que la plupart des requêtes utilisateurs sont

brièvement formulées [12][13], ce qui ne donne au moteur de recherche aucun indice sur le sujet spécifique. Dans cette section, nous présentons un aperçu sur les interfaces de présentation de résultats de recherche les plus courantes.

2.3.1 Liste ordonnée

La liste ordonnée est le mode de présentation de résultats de recherche le plus utilisé de nos jours. Dans ce modèle les documents récupérés en réponse à une requête sont triés selon la pertinence par rapport à la requête. Chaque élément de la liste se compose généralement du titre du document, son URL et un court extrait appelé « *Snippet*» (Figure 5).

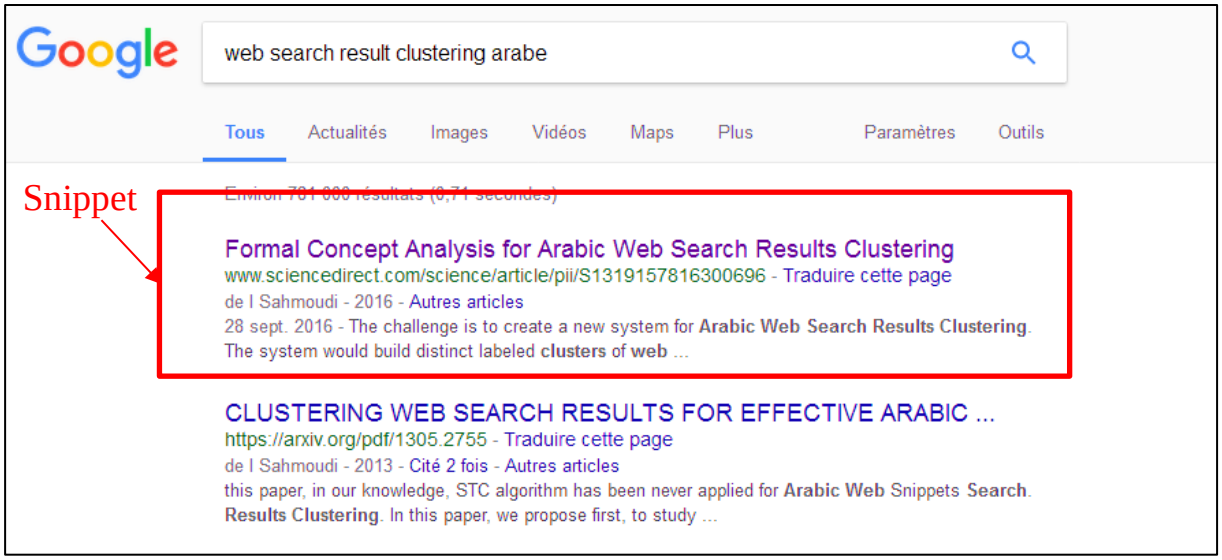


Figure 5 : *Exemple de présentation des résultats de recherche sous forme de liste ordonnée*

La présentation de la liste ordonnée est populaire et largement soutenue mais elle présente un certain nombre d'inconvénients sérieux [14]:

- Les utilisateurs doivent consulter une longue liste de documents, dont certains ne sont pas pertinents par rapport à son besoin.
- Aucune information explicite n'est fournie sur les relations entre les documents figurants sur la liste.
- Tous les documents de la liste doivent être triés même si certains d'entre eux ne se rapportent pas au même thème et ne sont pas comparables.

2.3.2 Répertoires web

Les répertoires Web sont créés en affectant manuellement des ressources internet aux branches d'un catalogue thématique hiérarchique (Figure 6). Le classement se fait typiquement dans une arborescence de catégories, censée couvrir tout ou partie des centres d'intérêt des visiteurs.

Avec le développement des moteurs de recherche tels qu'AltaVista et Google, les répertoires Web ont perdu de leur intérêt pour les utilisateurs, particulièrement en ce qui concerne les répertoires

généralistes. Ces derniers ont en effet beaucoup de mal à lutter face à la rapidité et la simplicité d'utilisation des moteurs de recherche modernes.

Aujourd'hui les annuaires web sont régulièrement utilisés dans une optique SEO (Search Engine Optimisation) par les référenceurs professionnels ou amateurs car ils permettent d'obtenir des hyperliens pointant vers leurs sites Web. Le fait d'obtenir ces liens donne de la valeur à ces sites du point de vue des moteurs de recherche, ce qui est susceptible de favoriser leur positionnement dans les résultats de recherche. Au-delà de cela, un répertoire Web moderne bien modéré et suivi permet tout de même de trouver des sites de qualité puisqu'ils ont été acceptés au sein de l'annuaire après une vérification humaine, contrairement aux moteurs qui eux sont basés sur un algorithme.

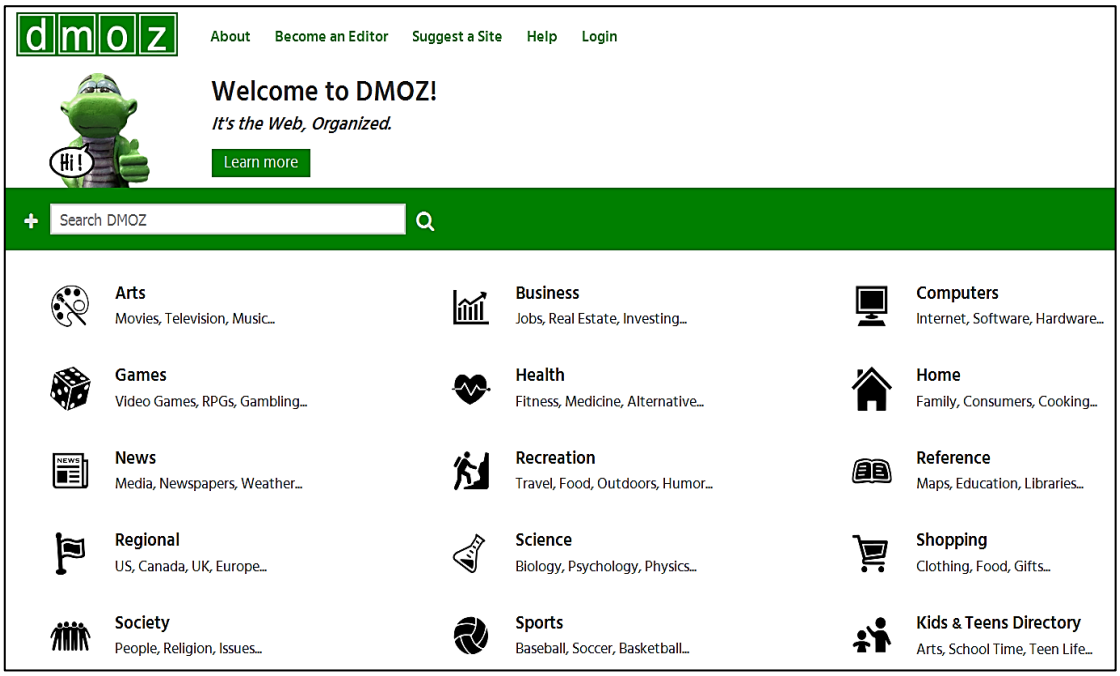


Figure 6 : *Exemple de répertoires Web (Open Directory Project)*

2.3.3 Le regroupement des résultats : Représentation thématique

La représentation thématique produite par le regroupement des résultats de la recherche retournés en réponse à une requête consiste à regroupe les documents similaires dans un groupe cohérent. Il est souvent plus facile de balayer quelques groupes cohérents que de nombreux documents individuels. Ceci est particulièrement utile si les mots cherchés ont différents sens. L'exemple de la figure 7 est «*jaguar*». Trois sens fréquents sur le Web font référence à :

- La voiture
- L'animal
- Système d'Exploitation Apple.

Le panneau des résultats retournés et regroupés par le moteur de recherche *Vivísimo (http://vivisimo.com)* peut être une interface utilisateur plus efficace pour comprendre ce qui est dans les résultats de recherche plutôt qu'une simple liste de documents.

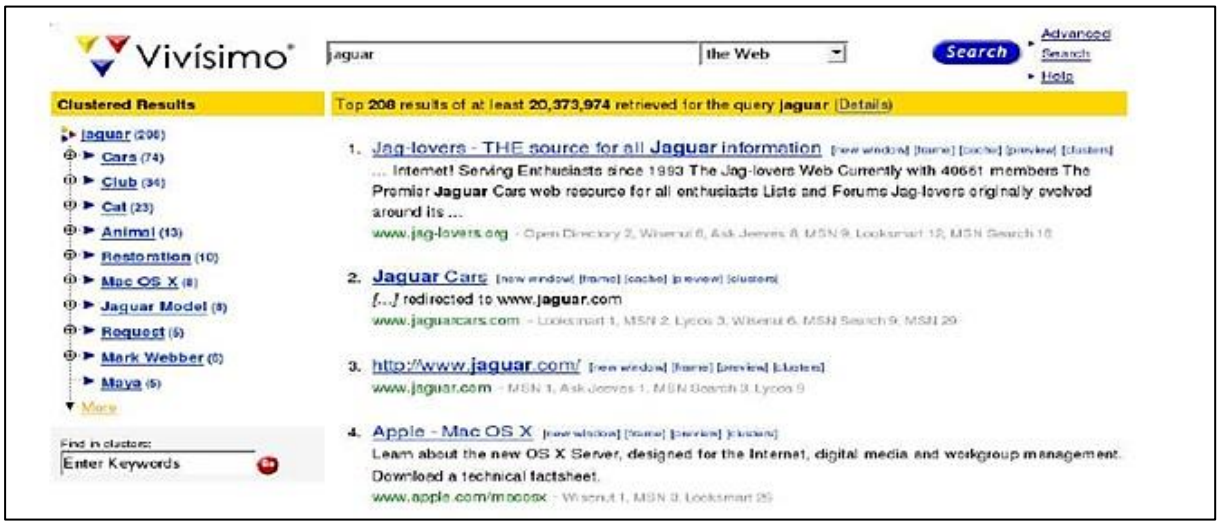


Figure 7 : *Résultats de recherche en utilisant Vivísimo*

3 Le clustering des résultats d'un SRI

3.1 État de l'art du clustering appliqué à la recherche d'information

Le clustering des résultats d'un SRI a été introduit pour la première fois par Cutting et al. en 1992. Les clusters y étaient générés en utilisant deux algorithmes de clustering introduit avec le système *Scatter/Gather* [22]:

- le *Buckshot* qui cherche à améliorer les algorithmes de type k-means en sélectionnant $\sqrt{k.n}$ graines au lieu de k puis les fusionnent en k graines ;
- la *Fractionalisation* qui répartit le corpus en sous-corpus de tailles réduites,qui réalise une agglomération des corpus pour en réduire le nombre puis classifie le corpus réduit.

Son principal avantage résulte de sa complexité linéaire. En revanche, il ne permet pas de traiter le problème de la polysémie, chaque document étant associé à un seul cluster. De plus, le nombre de clusters produits est un paramètre fixé par l'utilisateur, ce qui nous semble peu réaliste. Ce nombre de clusters ne doit-il pas plutôt dépendre de la polysémie de la requête ? N'existe-t-il pas des requêtes pour lesquelles une organisation des résultats en plusieurs groupes n'a aucun sens ? Nous pensons que ce sont là des points importants qui ont souvent été mis de côté dans les techniques proposées dans l'état de l'art. Il s'agit d'un algorithme centré sur les données (data-centric[2]), où l'étiquetage des clusters n'est pas forcément pertinent et interprétable par l'utilisateur (pour chaque cluster, les termes les plus fréquents sont utilisés comme étiquettes) [15].

Une approche très différente a ensuite été proposée par Zamir et al. dans le cadre du système *Grouper* [14]. Cette approche repose sur l'algorithme STC (Suffix Tree Clustering) qui rassemble les

[2] Selon la dénomination de Carpineto et al. (2009), les approches de clustering data-centric sont centrées sur la qualité des clusters ne se préoccupant pas de l'interprétabilité des clusters, alors qu'au contraire les méthodes description-aware sont centrées sur l'interprétabilité des clusters par l'utilisateur.

documents sur la base d'une seule propriété bien choisie : STC regroupe les documents qui partagent des expressions suffisamment fréquentes (dans l'ensemble des extraits de documents). L'astuce principale de STC repose sur l'utilisation d'un arbre de suffixe généralisé pour rechercher les séquences de termes les plus fréquentes en un temps linéaire. L'intérêt d'une telle approche est que les clusters ont un label clairement lisible pour l'utilisateur final – *description-aware* selon [15]. En revanche, nous doutons de la robustesse de telles méthodes, il suffit en effet de changer l'ordre de deux mots ou de remplacer un mot par un synonyme pour que les séquences les plus fréquentes changent complètement. Depuis ces deux travaux de référence, de nombreuses autres contributions ont été proposées, dont : *SHOC* [16], *SnakeT* [17], *Lingo* [18] et *Noodles* [19]. Il est assez remarquable que la plupart de ces méthodes (toutes celles citées ici sauf *SnakeT*) se basent sur la technique de *Latent Sementic Indexing* (LSI) introduite en 1990 par Deerwester et al. [20]. C'est compréhensible car, en prenant en considération la cooccurrence des termes, la *LSI* répond au problème de la synonymie. Egalement, la complexité importante de cette méthode est limitée par la taille des données réduites au sous-ensemble de documents restitués par un SRI.

A l'exception de *Scatter/Gather* [21] et *Noodles* [19], les autres méthodes sont basées sur les extraits de documents « *snippet* » plutôt que sur le texte complet. C'est un intérêt évident au regard de la complexité, et cela permet de faire facilement fonctionner les systèmes au-dessus d'un SRI existant en analysant uniquement (et en ligne) les snippets restitués par celui-ci. Par ailleurs, Zamir et al. [14] n'ont pas observé d'augmentation de la qualité des clusters en utilisant les documents complets plutôt que les snippets. Nous pouvons cependant penser que ce résultat est dû à un comportement particulier de l'algorithme STC. En effet, Mecca et al. [19] en 2007 montrent qu'avec une méthode basée sur la LSI, les résultats sont significativement meilleurs quand les documents complets sont utilisés.

Alors que les approches précédentes reposent principalement sur des modélisations vectorielles des documents, quelques auteurs ont traité le problème de clustering des résultats d'un SRI comme un problème de partitionnement de graphe [22], [23]. Le partitionnement d'un graphe consiste à trouver une partition de l'ensemble des sommets qui minimalise la taille de la coupe induite (c'est-à-dire le nombre total d'arêtes entre deux groupes). Newman [24] a montré que cela ne conduit pas forcément à un découpage sémantiquement intéressant du graphe. Le travail de Chen et al. [25] exploite une technique de détection de communautés (par optimisation de la modularité) sur un graphe de termes construit à partir des cooccurrences de ceux-ci dans les extraits de documents restitués. Le but est de trouver les « *word sense communities* » qui permettent ensuite de construire des clusters de documents. Comme le signalent les auteurs, la principale originalité de cette approche par détection de communautés est que le nombre de clusters n'est pas un paramètre fixé mais dépend des données. Nous conservons cet avantage. Les méthodes citées soit construisent des clusters de documents puis les étiquettent, soit construisent des clusters d'étiquettes puis leur attribuent des documents. Seules Dhillon

[23] et l'approche (non encore citée) par une analyse formelle de concept Carpineto [26] ne séparent pas le clustering de l'étiquetage (ou le clustering de « l'assignation »).

3.2 Moteurs de recherche avec visualisation thématique

Les Moteurs de recherche avec visualisation thématique présentent un grand intérêt pour la recherche d'information sur Internet, en offrant la possibilité de structurer en classes non prédéfinies à l'avance les résultats fournis par les moteurs de recherche. En effet, les nombreuses réponses fournies par les moteurs de recherche sont rarement structurées et placent donc l'utilisateur devant le problème d'analyse des résultats proposés. Les outils de regroupement ont donc été appliqués pour trouver des classes au sein de ces résultats.

3.2.1 Yippy

Yippy[3] est un métamoteur qui présente ses résultats de recherche classés par dossiers, eux-mêmes triés par pertinence de recherche. Il est la partie "publique" de Vivisimo, une solution de clustering créée à l'Université Carnegie-Mellon qui est aujourd'hui propriété d'IBM. La figure 8 présente l'interface web de Yippy.

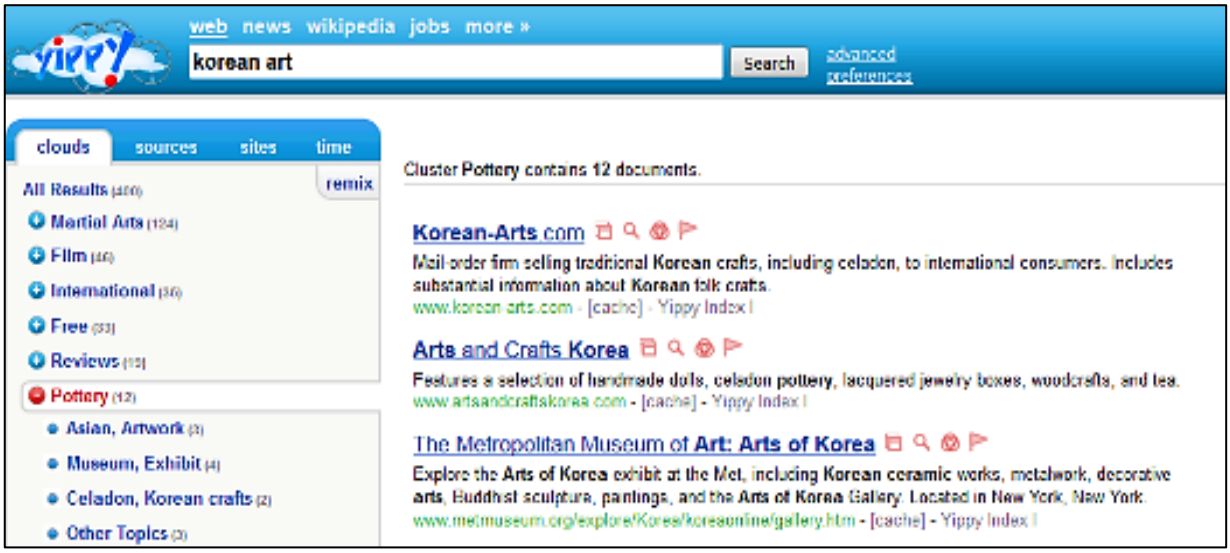


Figure 8 : *Interface de Yippy*

3.2.2 Hulbee

Hulbee[4], également appelé Swisscows, est un moteur de recherche suisse lancé en juin 2014 et développé par la société Hulbee AG. La figure 9 présente l'interface web de Hulbee.

[3] https://yippy.com/
[4] https://swisscows.com/?region=en-US

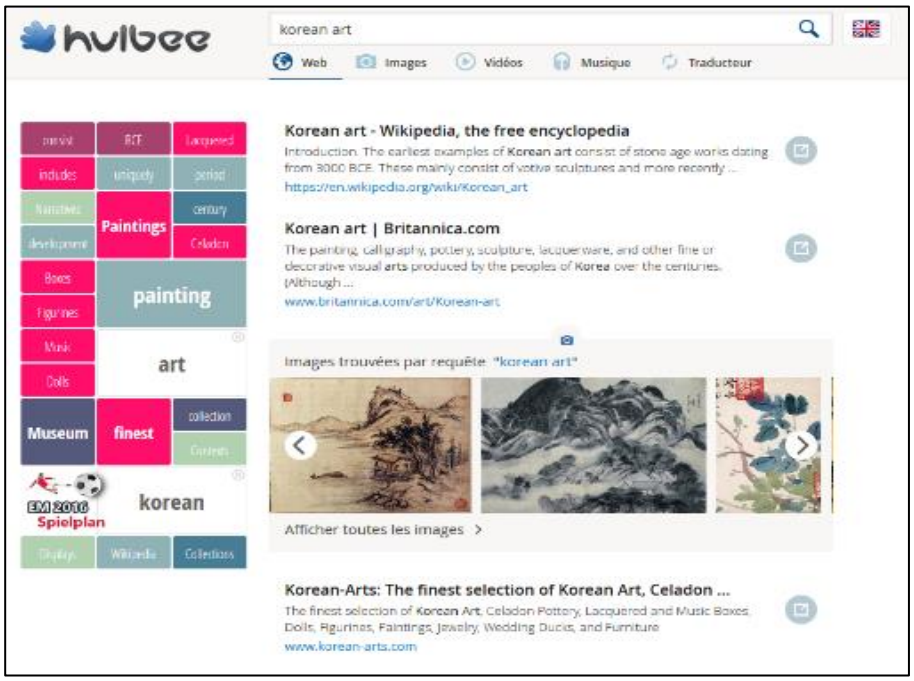


Figure 9 : *Interface de Hulbee*

3.2.3 Carrot²

Carrot² est une plateforme open source, incluant différents algorithmes de regroupement tels que lingo et suffixe tree. Elle permet aux utilisateurs web d'effectuer des recherches web. Les chercheurs peuvent aussi effectuer des tests et des évaluations en vue de mener des études comparatives. La figure 10 présente différents interfaces web de Carrot².

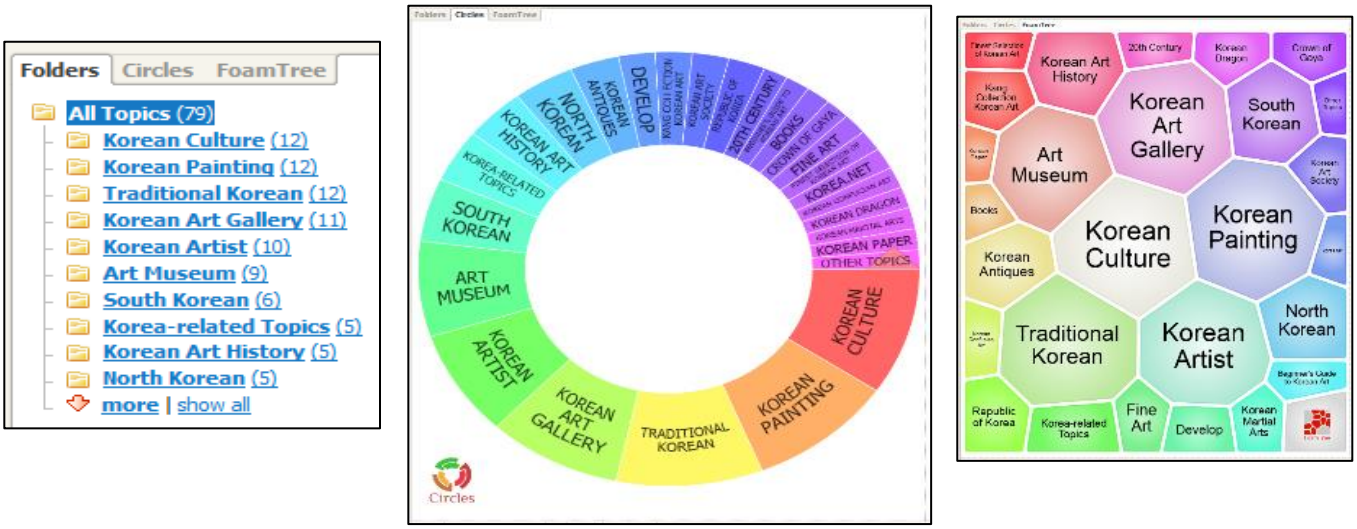


Figure 10 : *Interface de Carrot²*

4 Conclusion

Les moteurs de recherche sont les outils utilisés pour récupérer des informations à partir du Web. La façon dont les résultats de recherche sont présentés influence le processus de consultation, ce qui peut empêcher l'utilisateur à trouver sa recherche. Le clustring des résultats de recherche regroupe ces derniers en sous thématiques, cette façon de faire est plus avantageuse que la façon classique. Cependant la majorité des systèmes présentés dans ce chapitre sont spécifiques à d'autres langues autres que la langue arabe. Pour cela l'objectif des chapitres suivants est de présenter nos contributions que nous apportons à ce sujet de consultation pour la langue arabe.

CHAPITRE 2. SRI et la langue arabe

1 Introduction

Avec 78% de part de marché en décembre 2017[5] (Figure 11), Google est probablement le moteur de recherche le plus puissant sur le marché, ou du moins, le plus utilisé, car il existe bien une corrélation entre les deux constats.

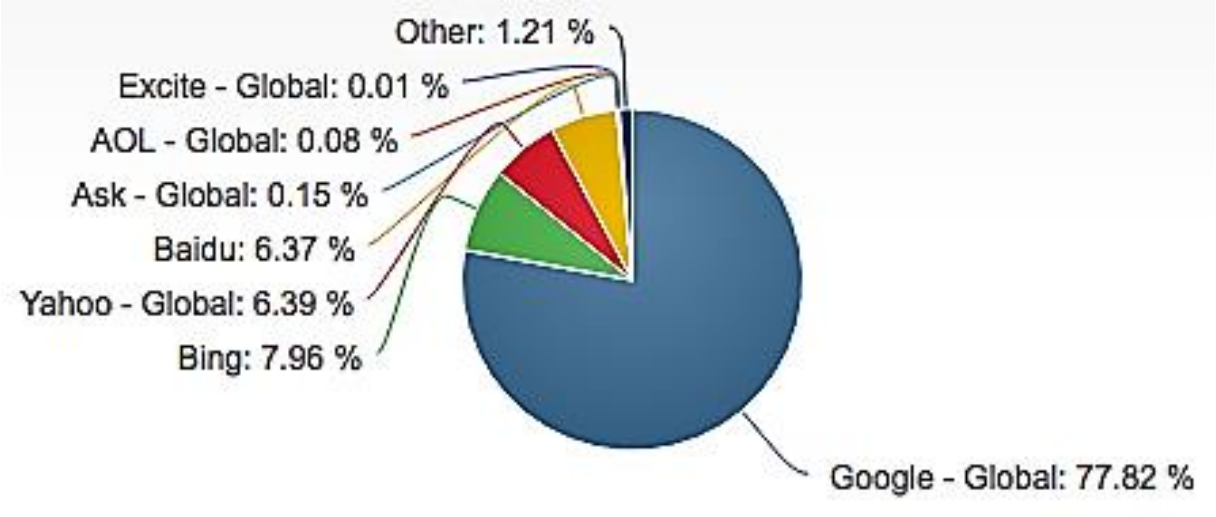


Figure 11 : *Top des moteurs de recherche en parts de visites*

Le moteur de recherche web Google reste très sensible au comportement des internautes. En effet, les algorithmes utilisés par Google pondèrent positivement ou négativement les pages en fonction du nombre de clics sur les liens retournés. L'algorithme ***PageRank*** de Google [27] note les pages en fonction des liens hypertextes qu'elles contiennent [25]. Selon que les pages aient été visitées ou non après une recherche, elles gagnent ou perdent des points dans le classement par rapport aux mots clés recherchés. Dans d'autres cas, selon que la page ait servi de routeur aux internautes ou pas (c'est-à-dire que la page ait été pertinente pour la recherche et aurait contenu des liens qui auraient été suivis par l'internaute), elle gagne ou perd des points dans les priorités d'indexation.

Le référencement pour les moteurs de recherche, ou SEO « *search engine optimization* », est un ensemble de techniques pour optimiser la visibilité d'une page web dans les pages de résultats de recherche. Ces techniques cherchent à améliorer la compréhension par les robots d'indexation de la thématique et du contenu d'une ou de l'ensemble des pages d'un site web et à augmenter le trafic naturel du site. L'objectif est d'améliorer le positionnement d'une page web dans les pages de résultats de recherche sur des mots-clés ciblés (selon les thèmes principaux du site). On considère que le positionnement d'un site est bon lorsqu'il est positionné (classé) dans la première page des résultats de recherche, dans l'une des dix premières réponses naturelles d'une recherche sur des mots-clés correspondant précisément à sa thématique.

[5] https://fr.semrush.com/blog/50-faits-incontournables-pour-votre-strategie-seo/

En effet, il est évident que, parmi les milliards de pages sur internet, les informations recherchées ne peuvent pas se retrouver uniquement sur une dizaine (ou quelques dizaines) de documents proposés.

Ce constat est encore plus réel lorsque Google est utilisé en langue arabe. En effet, la plupart des réponses figurant en tête de liste, pour une requête en arabe, proviennent des sites commerciaux bien référencés. L'algorithme ***PageRank*** de Google surclasse ces sources d'informations riches en renvois hypertextes et beaucoup plus utilisées dans le monde arabe au détriment d'autres sources d'informations en provenance de travaux universitaires, journalistiques, etc.

L'essentiel de la population des internautes arabophones maîtrise une seconde langue, le français ou l'anglais. Le web étant très bien développé dans l'une et l'autre de ces deux langues, l'internaute arabe s'oriente vers le multilinguisme pour une grande partie de ses recherches. Par certains aspects de spécialisation de la recherche, comme Google Scholar [28], le moteur de recherche donne souvent des réponses satisfaisantes. La mesure du degré de pertinence suppose une connaissance de l'ensemble des documents existants sur le web.

Dans le cadre de ce chapitre, pour remédier à ces problèmes, nous nous intéressons à un certain nombre de caractéristiques morphologiques et linguistiques afin de mettre en évidence la nécessité de traitements plus approfondis lors de l'indexation des documents écrits en langue arabe. Dans un premier temps, nous parlerons de la couverture des moteurs de recherche pour l'arabe et de la dissymétrie entre l'indexation et la recherche d'information, particulièrement palpable pour l'arabe non vocalisé. Par la suite, nous présentons l'apport des traitements linguistiques pour l'amélioration de la recherche en arabe.

2 La langue arabe sur le web

2.1 Couverture des moteurs de recherche web

Les robots des moteurs de recherche (spiders ou crawlers) parcourent les sites de la toile, à intervalles réguliers. L'exploration est indépendante de l'alphabet, elle dépend surtout des performances en terme de couverture de chacun des moteurs. La problématique de recherche d'information dépend de deux facteurs, le premier concerne l'indexation des pages et le second est lié à la recherche dans les index ou dans les pages elles-mêmes.

L'indexation des pages web se fait, pour l'essentiel des moteurs de recherche, par l'une ou la combinaison des méthodes suivantes [29] :

- La récupération des balises « méta » contenant les mots clés décrivant le contenu des pages et proposés par le créateur du site.
- La récupération du contenu de la balise « titre », il est d'ailleurs recommandé de donner des titres différents à chacune des pages web du site pour avoir un maximum de chance de ressortir ces pages parmi les résultats du moteur.

Pour les ressources jugées importantes, les robots peuvent indexer tout le contenu de la page. La recherche est la partie secrète des robots, les algorithmes pondèrent les pages en fonction d'un ensemble de critères comme la position du mot dans la page (titre, paragraphe, lien hypertexte) en fonction de l'historique ou encore de la nature de la ressource.

Les webmasters, de leurs côtés, positionnent les mots pertinents pour leurs sites dans les endroits stratégiques pour le robot de recherche. Cet aspect n'a pas de pertinence par rapport aux langues utilisées, mais il est pertinent selon le traitement linguistique possible dans chacun des moteurs, en fonction de leur maîtrise de la lemmatisation, de la dérivation ou de leurs ontologies pour relier des mots de la requête d'utilisateur aux mots clés proches sémantiquement ou faisant partie de la même famille morphologique.

2.2 Dissymétrie de l'indexation et la recherche en langue arabe

La recherche d'information en langue arabe montre souvent une certaine dissymétrie entre l'indexation et le traitement des requêtes provoquée particulièrement par l'absence des voyelles dans les textes arabes écrits et aussi par la nature agglutinante de l'écriture arabe.

Par exemple, lors de l'indexation d'un document, le verbe "écrire" (**كَتَبَ**), le nom "livres" (**كُتُب**) et le nom "écrit" (**كَتْب**) sont tous indexés sous une seule et même entrée (**كتب**, ktb), car ils ne sont généralement pas vocalisés dans le texte. Il en est de même pour le mot **علم** qui peut désigner plusieurs sens (drapeau, science, connaître, etc)[29].

Quelles que soient les précisions apportées à la recherche (même si on note le mot entièrement vocalisé), le moteur ne pourra pas séparer ces formes étant donné qu'elles ne sont pas vocalisées à la base. Par conséquent, les mots de l'index ne sont pas vocalisés non plus.

2.3 Recherche d'information en langue arabe

Une grande partie des requêtes utilisateurs sur le web, indépendamment des langues, concernent des entités nommées tels que des noms propres. R. Abbès [30] a fait un test sur un échantillon de 2850 requêtes arabes dans l'annuaire [31] qui lui permet de constater que 94,2% des requêtes concernent les formes nominales, 3,30% concernent les formes verbales et 2,5% concernent les mots grammaticaux. Bien entendu, ces valeurs peuvent être ajustées si nous prenons en compte le contexte non vocalisé des requêtes. En effet, en dehors de quelques formes verbales et de mots grammaticaux non ambigus comme "quand" متى nous retrouvons beaucoup de formes ambiguës comme نزل, nzl (descendre ou hôtel). Notons aussi que les formes verbales rencontrées ne sont pas fléchies.

La particularité des noms propres arabes est qu'ils sont souvent des dérivées de formes verbales (participe actif, participe passif, etc.). **كاتب ياسين** désigne à la fois l'écrivain et aussi un nom propre comme pour *Kateb Yassine*. Toutefois, la recherche par **كاتب** renvoie essentiellement écrivain. Nous constatons donc une grande faiblesse dans le traitement réservé aux entités nommées.

La recherche du mot "écrire" كَتَبَ (vocalisé) sur Google donne, sur les premiers résultats, la concept de «livres». S'agit-il d'une question de «ranking» ou de priorité donné aux noms ? Cependant, nous constatons que la voyellation du mot-clés n'a aucune influence sur la recherche.

Sauf langue anglaise, vocalisé ou non, Google utilise le modèle statistique, il ne fait pas la différence entre les langues et traite toutes les langues de la même manière.

3 Description et caractéristiques de la langue arabe

La langue arabe est parlée dans plus de *22* pays, du *Maroc* jusqu'à *l'Iraq* et dans toute la péninsule arabe. C'est la langue maternelle pour plus de *206* millions de personnes et la langue parlée pour plus de *300* millions de personnes[6]. L'arabe, langue du *Saint Coran*, est devenue la langue d'une civilisation et ne sert plus seulement à désigner les seuls habitants de la péninsule arabe qui la parlaient [32]. La richesse de la langue arabe dans laquelle se déploient des variétés écrites et orales, rend cette dernière capable de répondre à un spectre très diversifié d'usages sociaux [33]. Mais au-delà de cette diversité, les sociétés arabes ont une conscience aiguë d'appartenir à une communauté linguistique homogène. Elles sont attachées à l'intégrité de leur langue, d'où l'importance de *l'Arabe Standard Moderne* (*ASM)* qui constitue le terrain commun pour cette large arabe.

D'après *Boulaknadel* [33], par ses propriétés morphologiques et syntaxiques, le traitement automatique de la langue arabe doit faire face à :

- la nature agglutinante de la langue : l'ensemble des morphèmes collés à l'unité lexicale qui véhiculent plusieurs informations morphosyntaxiques ;
- la richesse flexionnelle de l'arabe ;
- l'absence de voyellation de la majorité des textes arabes écrits : ce phénomène entraîne un nombre important d'ambigüités morphologiques. En arabe, chaque lettre doit prendre un signe de voyellation et de surcroît les voyelles sont porteuses de certains traits morphosyntaxiques comme la déclinaison, le mode et le cas.

En outre des propriétés linguistiques, l'arabe recense un nombre de ressources linguistiques comprenant des lexiques monolingues et multilingues ainsi que des corpus de langue générale et des corpus de spécialité consacrés à une situation de communication ou à un domaine de la connaissance [33]. L'Arabe compte aussi un certain nombre d'outils linguistiques à savoir les analyseurs morphologiques ainsi que les extracteurs des racines basés essentiellement sur une procédure de désuffixation qui consiste à supprimer les suffixes qui différencient les flexions des unités lexicales (les formes conjuguées d'un verbe par exemple) [33].

L'arabe peut être considérée comme un terme générique rassemblant plusieurs variétés [33]:

[6] https://fr.statista.com/statistiques/565467/langues-les-plus-parlees-dans-le-monde/

- *l'Arabe Classique*[7] *:* est une des formes de l'arabe utilisée au moyen âge dans les textes littéraires pendant le califat omeyyade et abbasside (entre les VIIe et IXe siècles). Elle évolue au fil du temps de l'arabe précoranique à l'arabe coranique, puis à l'arabe post-coranique auquel est parfois réservée l'appellation « *arabe classique* ».
- *l'Arabe Standard Moderne (l'ASM) :* une forme un peu différenciée de l'arabe classique, et qui constitue la langue écrite de tous les pays arabophones. *L'ASM* reste le langage de la presse, de la littérature et de la correspondance formelle, alors que l'arabe classique appartient au domaine religieux et est pratiqué par les membres du clergé ;
- *les Dialectes Arabes :* malgré l'existence d'une langue commune, chaque pays a développé son propre dialecte. Issus de l'arabe classique, leurs systèmes grammaticaux respectifs affichent de nettes divergences avec celui de *l'ASM*. *Boulaknadel* [33] regroupe ces dialectes en quatre grands groupes :
    1. *les Dialectes Arabes*, parlés dans la Péninsule Arabique : dialectes du Golfe, dialecte du najd, yéménite ;
    2. *les Dialectes Maghrébins :* algérien, marocain, tunisien, hassaniya de Mauritanie;
    3. *les Dialectes proche-orientaux :* égyptien, soudanais, syro-libano-palestinien, irakien (nord et sud) ;
    4. *la langue Maltaise* est également considérée comme un dialecte arabe.

3.1 Particularités de la langue arabe

3.1.1 Alphabet Arabe

L'alphabet arabe comporte *28* consonnes (tableau 6) qui correspondent chacune à un phonème, *14* consonnes lunaires qui n'assimilent pas le ل de l'article et *14* consonnes solaires qui assimilent le ل de l'article.

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

Tableau 6 : *Consonnes arabes*

De même, il existe *six* voyelles en arabe standard (tableau 7), *trois* longues et *trois* courtes, la durée phonétique d'une voyelle longue étant environ le double de celle d'une voyelle courte [34].

Voyelles longues	Voyelles courtes
ا و ي	َ ُ ِ

Tableau 7 : *les voyelles*

[7] http://dictionnaire.sensagent.leparisien.fr/Arabe%20classique/fr-fr/

Les lettres arabes changent de forme selon leur position dans le mot. Toutes les lettres de l'alphabet (à l'exception de ء), qu'elles soient manuscrites ou imprimées, sont agglutinées soit au début soit au milieu soit à la fin. Cependant, ***six*** d'entre elles s'attachent uniquement aux lettres précédentes mais pas aux lettres suivantes (tableau 8) [34].

Les caractères qui ne s'attachent pas au suivant
ا د ذ ر ز و

Tableau 8 : *Caractères qui ne s'attachent pas au suivant*

La lettre «hamza-أ » peut également prendre ces formes : «أ،إ،آ،لأ،لإ», également pour la lettre «wāw-و» qui peut prendre cette forme «ؤ» [34].

3.1.2 Structure d'un mot Arabe

En arabe un mot peut signifier toute une phrase grâce à sa structure composée qui est une agglutination d'éléments de la grammaire [35]. La structure du mot arabe est donc décomposable en cinq éléments qui sont des morphèmes : ***proclitique, préfixe, base, suffixe*** et ***enclitique*** qui donnent des informations et des traits grammaticaux.

Les proclitiques et les enclitiques s'attachent au mot minimal par une seconde dérivation externe pour former le mot maximal. Par proclitique, nous désignons les proclitiques simples (morphèmes à une lettre) et les proclitiques composés (morphèmes à plusieurs lettres). Les premiers sont des coordonnants, des conjonctions, des prépositions, etc. Les seconds sont obtenus par combinaison des premiers. L'article (ال : al) est également considéré comme proclitique simple, bien qu'il comporte deux lettres. L'enclitique est un pronom personnel complément qui peut être simple ou double. Il est attaché au mot qui le précède pour ne former qu'un seul mot.

La représentation suivante schématise la structure possible du mot «أتعاقبوننا» qui exprime la phrase suivante en français : *«Est-ce que vous nous punissez ?»*. Il se décompose comme suit [34] :

- *proclitique* أ : conjonction d'interrogation ;
- *préfixe* ت : préfixe verbal du temps de l'inaccompli ;
- *base* عاقب : corps schématique dérivé de la racine ع ق ب;
- *suffixe* ون : suffixe verbal exprimant le pluriel ;
- *enclitique* نا : pronom, suffixe complément du nom.

a) Préfixes

Le préfixe est un morphème verbal relatif à l'inaccompli (actif ou passif) placé avant la base. Pour les noms et les autres aspects de la conjugaison nous dirons que le préfixe est vide. Nous prenons la racine à la 3ème personne du singulier de l'accompli, par exemple : كَتَبَ. A cette forme nous ajoutons un pronom personnel préfixe, par exemple « il » ي, le verbe passe alors au cas nominatif : يَكْتُبُ.

b) Suffixes

Le suffixe est un morphème situé immédiatement après la base. La base peut être une base verbale suivie d'un suffixe verbal ou une base nominale suivie d'un suffixe nominal. Un suffixe peut être une combinaison de deux suffixes tel que (ياء النسبة : Yâ ALNiSBa).

3.1.3 Morphologie arabe

Le lexique arabe comprend *trois* catégories de mots : verbes, noms et particules. Les verbes et les noms sont le plus souvent dérivés d'une racine à trois consonnes radicales [36]. Une famille de mots peut ainsi être générée d'un même concept sémantique à partir d'une seule racine à l'aide de différents schèmes. Ce phénomène est une caractéristique de la morphologie arabe. On dit donc que l'arabe est une langue à racines réelles à partir desquelles on déduit le lexique arabe selon des schèmes qui sont des adjonctions et des manipulations de la racine. On peut dériver un grand nombre de noms, de formes et de temps verbaux.

La majorité des verbes arabes ont une racine composée de *trois* consonnes. L'arabe comprend environ *150* schèmes ou patrons dont certains plus complexes, tel le redoublement d'une consonne ou l'allongement d'une voyelle de la racine, l'adjonction d'un ou de plusieurs éléments ou la combinaison des deux. Une autre caractéristique est le caractère flexionnel des mots : les terminaisons permettent de distinguer le mode des verbes et le cas des noms [36].

3.1.4 Catégories des mots arabes

La langue arabe comprend trois catégories de mots : le verbe, le nom et le particule [34]:

- le verbe est une entité qui exprime un sens dépendant de l'un des trois temps, l'accompli, l'inaccompli et l'impératif. Les verbes se divisent en deux catégories, les verbes *«sains»* et les verbes *«défectueux»* :
  - ✓ les verbes *sains* ne contiennent aucune des lettres défectueuses dans leur radical, à savoir les trois voyelles *«alif-*ا», «yāʾ-ي», «wāw-و».
  - ✓ les verbes *défectueux* contiennent au moins une lettre défectueuse.
- le nom désigne un objet ou un être exprimant une notion indépendante du temps ;
- les particules servent à lier les noms, les verbes et les parties de la phrase mais également à indiquer le temps.

3.1.5 Grammaire et caractéristiques de la langue arabe

La grammaire traditionnelle se divise en deux branches [33] :

1. La morphologie, الصرف, qui comprend :
   - *Morphologie dérivationnelle*, qui étudie la construction des unités lexicales et leur transformation selon le sens voulu. Ainsi, la dérivation morphologique est décrite sur

une base morphosémantique : d'une même racine, se dérivent différentes unités lexicales selon des schèmes qui sont des adjonctions et des manipulations de la racine.

- *Morphologie flexionnelle* concerne le marquage casuel pour le nom et l'adjectif ou la conjugaison du verbe, appelé "الإعراب".

2. La syntaxe, "النحو ", qui étudie la construction correcte des phrases garantit la grammaticalité de la phrase en analysant :
   - la position des unités lexicales les unes par rapport aux autres, déterminant ainsi l'ordre des unités lexicales ;
   - le marquage casuel des unités lexicales de la phrase. Ainsi, la fonction syntaxique de l'unité lexicale est déterminée en s'appuyant sur la morphophonologie.

3.2 Difficultés de l'analyse automatique de la langue arabe

3.2.1 Complexités du traitement automatique de la langue arabe

La langue arabe est une langue difficile pour un certain nombre de raisons [37] :

- L'orthographique avec des signes diacritiques est moins ambiguë et plus phonétique en arabe, certaines combinaisons de caractères peuvent être écrites de différentes façons.
- La langue arabe a des voyelles courtes qui aident à des prononciations différentes. Grammaticalement ils sont nécessaires, mais omis dans les textes arabes écrits.
- Manque des corpus arabes librement accessibles.

3.2.2 Exemples qui montrent la complexité de la langue arabe

a) Sens des mots

Le sens d'un mot d'après son contexte, le tableau 9 présente un exemple pour le mot قلب.

Le sens du mot	La phrase
L'actualité importante	في قلب الأحداث
Le cœur	أجرى عملية قلب مفتوح
Le cœur du stade	الكرة في قلب الملعب

Tableau 9 *: Différents sens du mot (قلب)*

b) Variations dans la catégorie lexicale

Le sens d'un mot d'après son catégorie lexicale, le tableau 10 présente un exemple pour le mot ذهب.

Le sens du mot	La catégorie du mot	Phrase
Or	Nom	ذهب خالص
Aller	Verbe	ذهب إلى المدرسة

Tableau 10 : *Catégorie lexicale du mot (ذهب)*

c) Synonymes

Les langues ont beaucoup de mots qui sont considérés comme des synonymes. Grâce à un corpus donné, les chercheurs peuvent utiliser les outils de l'analyse morphologique pour connaitre les synonymes d'un mot, la fréquence de chaque mot de ces synonymes et le plus connu entre eux. Des

exemples des synonymes en arabe sont *(وهب أعطى، منح، بذل،)* qui signifie *(donner)*, *(عائلة, أسرة)* qui signifie *(famille)*, et *(صف, فصل)* qui signifie *(classe)* [37].

d) Forme de mot en fonction de son cas

La forme de quelques mots arabes peut changer en fonction de leurs modes de cas *(nominatif, accusatif ou génitif)*. Par exemple le pluriel du mot *(*مسافر*)* qui signifie (voyageur) peut avoir la forme *(*مسافرون*)* dans le cas du nominative *(*مرفوعة*)* et la forme *(*مسافرين*)* dans le cas de *l'accusatif / génitif (*مجرورة / منصوبة*)* [37].

e) Caractéristiques morphologiques

Un mot arabe peut être composé d'un lemme, des affixes et des clitiques. Le lemme est obtenu à partir d'une racine consonantique *(*جذر*)* et d'un modèle. Les affixes flexionnels comprennent des marqueurs *(*علامات أو حركات إعرابية*)* pour le temps, le sexe et/ou le nombre. Les clitiques comprennent certaines prépositions *(*حروف جر*)*, conjonctions *(*حروف العطف*)*, déterminants *(*محددات*)*, et pronoms *(*ضمائر*)*. Les clitiques attachés au début d'un stem sont appelés proclitique et ceux fixés à l'extrémité de celui-ci sont appelés enclitiques [37].

3.2.3 Problème d'encodage

La langue arabe a des problèmes d'affichage parce que son encodage diffère selon la plate-forme de la machine. La figure 12 montre le problème d'encodage où toutes les cellules ombrées sont affichées correctement tandis que les autres cellules ne sont pas correctes. Le prétraitement et la classification du texte arabe avec un encodage incorrect peut conduire à des résultats erronés. Le tableau 11 présente les caractéristiques des deux systèmes communs d'encodage arabes : *Unicode et CP-1256* [37] .

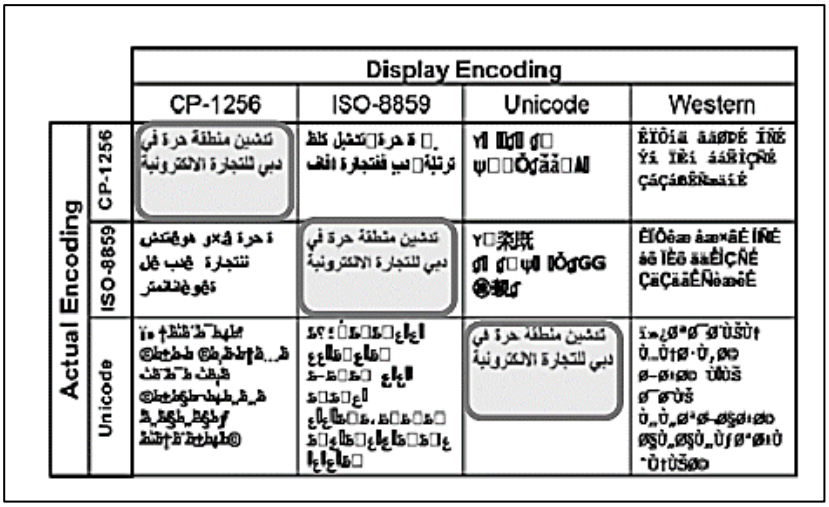


Figure 12 : *le problème de l'encodage de la langue arabe* [37]

Unicode	CP-1256 (code page 1256 Arabic windows)
Devenir la norme standard de plus en plus	couramment utilisé
Caractères codés sur 2 octets	Caractères codés sur 1 octet
Entrée / affichage largement soutenu	Entrée / affichage largement soutenu
Prise en charge étendue des caractères arabes	Ne peut pas supporter les longs caractères arabes
Représentation multi-script	Supporte bi-script (Roman / arabe)
Prise en charge des formulaires de présentation (formes et des ligatures)	Supporte trilingue : arabe, français, anglais (ANSI)

Tableau 11 : *L'encodage de la langue arabe (Unicode vs. CP-1256)* [37]

3.2.4 Analyse morphologique

L'analyse morphologique s'intéresse à la construction des mots à travers des processus de flexion (marques de genre, nombre, de conjugaison...), dérivation et composition. Étant donné un mot, il s'agit de déterminer quelles sont les unités minimales de sens qui le composent. Ces unités minimales de sens sont appelées morphèmes et se déclinent en termes de racine et d'affixes. Une analyse morphologique complète précise en plus la catégorie grammaticale de la racine et associe aux affixes des informations sémantiques et flexionnelles. Elle est réalisée soit en utilisant des bases lexicales existantes, soit à l'aide de véritables systèmes d'analyse, plus à même de traiter les formes non répertoriées.

Le problème principal de cette analyse réside dans l'agglutination et l'absence de voyellation. Pour l'agglutination en arabe et contrairement aux langues latines, les pronoms, les prépositions, les articles, les conjonctions, et autres particules collent aux noms, verbes, adjectifs et particules auxquels ils se rapportent. Comparé au français, un mot arabe peut parfois correspondre à une phrase française.

Cette caractéristique engendre une ambiguïté morphologique au cours de l'analyse. Ainsi, la reconnaissance des unités lexicales qui composent une unité morphologique n'est pas toujours facile à détecter. Le problème est de reconnaître que la bonne segmentation réside ainsi, dans la difficulté de distinction entre un proclitique ou enclitique et un caractère original du mot.

Par exemple, le caractère *"و"* dans le mot *« il est arrivé |* وصل *»* est un caractère original alors que dans le mot *"et il a ouvert |* وفتح *"*, il s'agit plutôt d'un proclitique.

L'absence de voyellation pose un autre problème important. En effet, les mots non voyellés engendrent beaucoup de cas ambigus au cours de l'analyse (e.g. le mot non voyellé *"*فصل*"* pris hors contexte peut être un verbe au passé conjugué à la troisième personne du singulier *"il a licencié |* فَصَلَ*"*, ou un nom masculin singulier *"chapitre/ saison |* فَصْلٌ*"*, ou encore une concaténation de la conjonction de coordination *"puis |* ف *"* avec le verbe *"*صل*"* : impératif du verbe lier conjugué à la deuxième personne du singulier masculin) [32] .

L'analyse morphologique est réalisée soit en utilisant des bases lexicales existantes, soit à l'aide de véritables systèmes d'analyse, plus à même de traiter les formes non répertoriées. Elle est présentée

dans de nombreuses applications du T*raitement Automatique des Langues (TAL*) et permet de reconnaître la présence d'un même mot sous des formes de surface différentes. L'une des analyses morphologiques les plus simples mises en œuvre en Recherche d'Information est la Racinisation. En extraction de connaissance, l'analyse morphologique la plus employée est la lemmatisation qui permet d'associer à une forme fléchie une forme conventionnelle ainsi que de calculer les traits flexionnels.

3.2.5 Segmentation de texte arabe

La segmentation d'un texte est une étape fondamentale pour son traitement automatique ; son rôle est de découper le texte en unités simples d'informations. En effet, l'opération de segmentation d'un texte consiste à délimiter les segments de ses éléments de base qui sont les caractères, en éléments constituants différents niveaux structurels tels que : paragraphe, phrase, syntagme, mot graphique, mot-forme, morphème, etc [32].

Pour la langue arabe, *Maâloul et al.* [32] affirment que les particularités de cette langue, rend la segmentation arabe toujours beaucoup plus difficile par rapport aux autres langues, étant donné qu'il n'y a pas de majuscules qui marquent le début d'une nouvelle phrase. De plus, les signes de ponctuation, ne sont pas utilisés de façon régulière.

D'après l'étude réalisée par *Belguith et al.* [38], certaines particules comme *"et* |و*"*, *"donc*| ف*"*, etc. jouent un rôle principal dans la séparation de phrases et peuvent être déterminantes pour guider la segmentation.

Pour la segmentation de texte *Ouersighni et al.* [39] utilisent :

- Une segmentation morphologique basée sur la ponctuation,
- Une segmentation basée sur la reconnaissance de marqueurs morphosyntaxiques ou des mots fonctionnels comme : حتى ، لكن ، أي ، و ، أو *(ou, et, mais, quand)*.

3.2.6 Etiquetage grammatical

L'étiquetage grammatical est l'opération qui consiste à attribuer à chacun des mots d'un texte la catégorie (non, verbe, adjectif, article défini, etc.) qui est la sienne dans le contexte où il apparaît [32].

Dans le même cadre, la difficulté de l'étiquetage grammatical [32] peut s'amplifier lorsque les textes visés se présentent sous leur forme non pas voyellée, mais partiellement voyellée ou encore totalement non voyellée, ce qui correspond au cas le plus courant.

Ce problème de voyellation pour un mot [32] est ainsi posé du fait que le choix de l'accentuation qui convient au mot est difficile et dépend essentiellement du contexte.

Le tableau 12 présente le problème d'ambiguïté grammaticale rencontrée lors de l'attribution catégorique d'un mot non voyéllé *"*كتب*"*, qui admet au moins quatre étiquettes grammaticales qui sont les suivantes [32]:

Exemple de voyellation	Étiquettes grammaticales
كُتُبٌ kutubun : des livres	substantif, masculin, pluriel
كَتَبَ kataba : il à écrit	verbe, 3ème personne masculin, singulier de l'accompli actif
كُتِبَ kutiba : il a été écrit	verbe, 3ème personne masculin, singulier de l'accompli passif
كَتِّبْ kattib : fais écrire	verbe à l'impératif, 2ème personne masculin, singulier

Tableau 12 : *Exemple d'étiquettes grammaticales attribuées selon la voyellation* [32]

3.2.7 Analyse syntaxique

Dans [32], *Maâloul* définit l'analyse syntaxique comme une analyse qui permet d'associer à un énoncé sa ou ses structures syntaxiques possibles, en identifiant ses différents constituants et les rôles que ces derniers entretiennent entre eux. Cette analyse syntaxique prend en entrée le résultat de l'analyse lexicale (éventuellement de l'étiquetage morpho-syntaxique) et fournit en sortie une structure hiérarchisée des groupements structurels et des relations fonctionnelles qui unissent les groupements.

Il a signalé aussi, dans la même voie de recherche, que les ambiguïtés vocaliques et grammaticales, relatives à la non voyellation des mots, posent des difficultés au niveau de l'analyse syntaxique. Ainsi, une phrase, en absence de la voyellation, peut être interprétée et traduite selon plusieurs interprétations qui sont toutes syntaxiquement correctes [32].

a) Absence de voyelles

L'absence de la voyellation dans les textes arabes présente un important problème lors de l'analyse automatique de ces textes. *Maâloul* et al. [32] affirment que l'ambiguïté grammaticale augmente si le mot est non voyellé. Cela est dû au fait qu'un mot non voyellé possède plusieurs voyellations possibles, et pour chaque voyellation est associée une liste différente de catégories grammaticales.

b) Agglutination

Dans la langue arabe, les articles, les prépositions, les pronoms, etc. se collent, contrairement aux langues latines, aux adjectifs, aux noms, aux verbes et aux particules auxquels ils se rapportent. Comparé au français, un mot en arabe peut parfois correspondre à une phrase en français. Cette caractéristique peut engendrer une ambiguïté au niveau morphologique pour cette langue. En effet, il est parfois difficile de distinguer entre un proclitique ou enclitique et un caractère original du mot [32].

c) Irrégularité de l'ordre des mots dans la phrase

Généralement l'ordre des mots dans une phrase écrite en langue arabe est relativement libre. Nous pouvons mettre au début de la phrase le mot sur lequel on veut attirer l'attention et l'on termine sur le terme le plus long ou le plus riche en sens ou en sonorité [32]. Cet ordre provoque des ambiguïtés

syntaxiques artificielles, dans la mesure où il faut prévoir dans la grammaire toutes les règles de combinaisons possibles d'inversion de l'ordre des mots dans la phrase [38].

Ainsi par exemple, le changement de l'ordre des mots dans la phrase ne produit pas deux phrases ayant des sens différents.

d) Absence de ponctuation régulière

Pour la langue arabe Standard Moderne, utilisée actuellement sur le web et les différents outils de médias sociaux, il y'a un manque de signes de ponctuations et des marqueurs typographiques ; et même dans le cas où ils y figurent, ils ne sont pas gérés par des règles précises d'utilisation.

Par ailleurs, les paragraphes écrits avec l'ASM ne contiennent aucun signe de ponctuation à part un point à la fin de ce paragraphe. Ainsi, il est à noter que la présence des signes de ponctuation ne peut pas guider la segmentation comme c'est le cas pour d'autres langues, telles que le français ou l'anglais. Ainsi, la segmentation de textes arabes doit être guidée non seulement par les signes de ponctuations et les marqueurs typographiques mais aussi par des particules et certains mots tels que les conjonctions de coordination, etc [38].

4 Conclusion

La recherche d'information en langue arabe se heurte inlassablement à ses spécificités linguistiques, notamment par sa morphologie d'une part riche et d'autre part complexe, par l'absence de signes de vocalisation dans les textes publiés et par la richesse des mots graphiques arabes. Ces spécificités sont souvent à l'origine d'une grande dissymétrie entre l'indexation et la recherche.

Dans ce chapitre, nous avons présenté une partie de la typologie des contraintes morphologiques de la langue arabe et montré le comportement des outils de recherche d'informations face à ces contraintes. Nous avons montré que la forme graphique des mots ne peut être utilisée pour la constitution de familles morphologiques, en raison de l'agglutination et de la richesse dérivationnelle en langue arabe.

Conclusion de la partie 1

Dans cette partie, nous avons présenté les problèmes liés aux moteurs de recherche. Le premier problème concerne la représentation des résultats de recherche, car la façon dont les résultats de recherche sont présentés influence négativement sur le processus de consultation, ce qui peut empêcher l'utilisateur à trouver sa recherche rapidement.

Le deuxième problème concerne l'indexation. Nous avons présenté la typologie des contraintes morphologiques de la langue arabe et montré le comportement des outils de recherche d'informations face à ces contraintes. Nous avons montré que la forme graphique des mots ne peut être utilisée pour la constitution de familles morphologiques, en raison de l'agglutination et de la richesse dérivationnelle en langue arabe.

L'objectif des deux parties suivantes (partie 2 et partie 3) consiste à présenter les contributions que nous avons apportées pour l'amélioration d'une part au niveau de processus de consultation (partie 2) et d'autre part au processus d'indexation (partie 3).

Partie 2 : Problème de Consultation

- **Chapitre 3 : L'arbre des suffixes pour le regroupement thématique des résultats de SRI**
- **Chapitre 4 : Analyse de concepts formels pour le regroupement thématique des résultats de SRI**

Introduction de la partie 2

Les statistiques internationales des utilisateurs d'internet par langue en 2017[8] (Figure 13), montrent une croissance impressionnante de l'arabe sur internet avec 219 millions d'utilisateurs. Ainsi que, le nombre de documents arabes disponibles sur Internet croît à un rythme très rapide. Par conséquent, aider les utilisateurs arabes à trouver une réponse à leurs besoins devient un sujet intéressant pour la recherche.

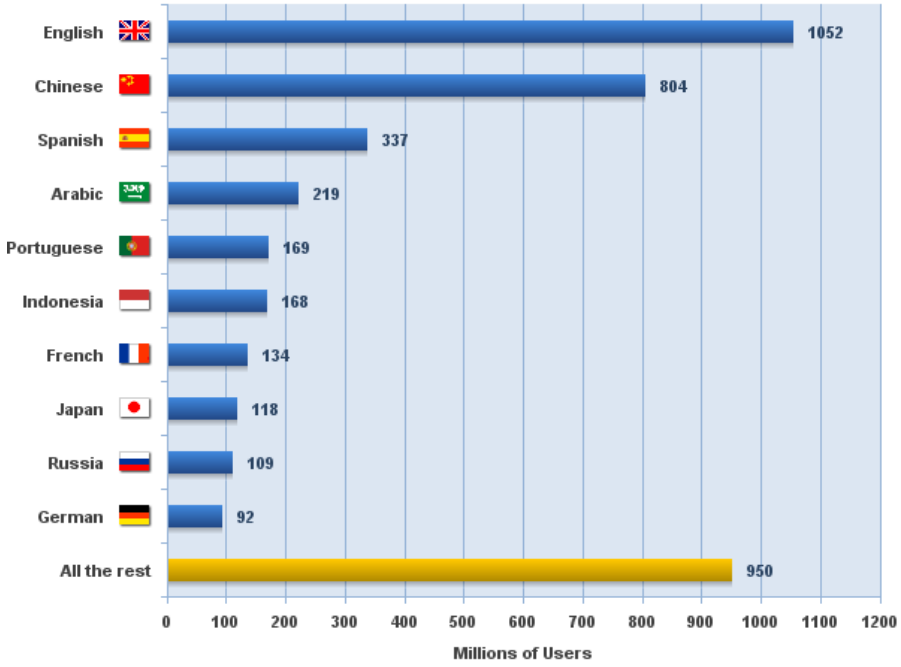


Figure 13 : *Statistiques internationales des utilisateurs d'Internet par langue en 2017*

Les moteurs de recherche sur le Web sont devenus indispensables dans la recherche de l'information pertinente du World Wide Web. Le nombre de documents en langue arabe disponibles sur Internet croit énormément. Pour répondre aux requêtes en langue arabe, les moteurs de recherche disponibles, tels que (Google[9], Yahoo[10] et Bing[11]) retournent une liste ordonnée des résultats de recherche (Snippets) qui contient un mélange de documents de la langue arabe et d'autres langues.

Même si, nous pouvons personnaliser la requête de l'utilisateur pour avoir des résultats seulement en langue arabe, la liste des URLs n'est pas assez efficace puisque le nombre de résultats de recherche récupérés peut être des milliers pour une requête typique.

En fait, le processus de consultation des résultats de recherche en utilisant une liste ordonnée comme résultat d'une requête d'utilisateur spécifique prend du temps, et le style de consultation ne semble pas convivial spécifiquement avec une requête ambiguë. Généralement, la plupart des utilisateurs face à une requête spécifique visualisent seulement les résultats affichés dans les premières

[8] https://www.internetworldstats.com/stats7.htm
[9] http://www.google.com
[10] http:// www.yahoo.com
[11] http://www.bing.com

pages, alors que des documents pertinents peuvent manquer. En outre, la plupart des documents arabes dans le Web ne contiennent aucune trace de signes diacritiques, ce qui augmente l'écart entre les besoins des utilisateurs et les résultats présentés dans les premières pages. Dans ce cas, le Web Search Results Clustering (WSRC) est d'une importance cruciale pour le regroupement de documents similaires afin d'améliorer et faciliter la consultation dans les pages Web sous une forme plus compacte et thématique. De nombreuses solutions commerciales ont été proposées ces dernières années, comme iBoogie, yippy, Kartoo[12,] Dogpile [13]. Cependant, ces solutions ont été développées spécialement pour les langues dont l'orthographe est basée sur le script latin ou utilisent un système de traduction de l'arabe vers l'anglais pour construire des clusters différents en utilisant une variété d'algorithmes de clustering. Le système doit construire des clusters étiquetés distincts à partir des snippets retournés par des moteurs de recherche pour répondre aux besoins des utilisateurs arabophones.

Cette partie comprend deux chapitres qui présentent deux contributions concernant la consultation. Dans le premier chapitre, nous proposons un nouveau système pour améliorer le processus de consultation pour la langue arabe. Ce système est basé sur les arbres de suffixe, qui sera interactif avec les internautes arabes afin de leur aider à trouver leurs besoins rapidement. Le deuxième chapitre présente une autre contribution qui se base principalement sur l'utilisation de l'analyse de concepts formels «*Formal Concept Analysis*» (FCA) comme un nouveau système pour le regroupement des résultats de recherche pour la langue arabe en fonction de leur structure hiérarchique.

[12] http://fr.kartoo.com/
[13] http://www.dogpile.com/

CHAPITRE 3. L’arbre des suffixes pour le regroupement thématique des résultats de SRI

1 Introduction

Suffix Tree Clustering (STC) est un algorithme de regroupement de résultats de recherche sur le Web basé sur une structure de données appelée arbre de suffixe. Le STC a été introduit et utilisé avec succès et abondamment avec différentes versions adaptées pour la langue latine [14] et la langue chinoise [40, 41].

Cependant, l'algorithme STC n'a jamais été appliqué à la langue arabe. Dans ce chapitre, nous commençons tout d'abord par étudier comment le STC peut être appliqué pour la langue arabe. En effet, nous avons constaté qu'il est impossible d'appliquer le STC après l'étape de prétraitement arabe (stem ou racine), vu que ce processus fusionne plusieurs clusters qui sont sémantiquement différents. Pour surmonter ce problème, nous proposons d'intégrer le STC dans un nouveau système en tenant compte des propriétés de la langue arabe afin de rendre le Web de plus en plus adapté aux utilisateurs arabes. Aussi, nous avons ajouté une étape qui a rendu le système interactif avec l'utilisateur pouvant l'aidé à spécifier son besoin avec précision.

2 Le système proposé AWSRC

Dans le figure 14, nous présentons l'architecture de notre première contribution pour le regroupement thématique de résultats retournés par un moteur de recherche. Ce système nommé « *Arabic Web Search Results Clustering* » (AWSRC), est illustré dans [42] et peut être décrit selon le scénario suivant :

1) L'utilisateur spécifie une requête en langue arabe en utilisant l'interface web de notre système. Cette requête est envoyée automatiquement au moteur de recherche google en utilisant l'API fournit par ce dernier. Les résultats retournés sont sous forme des snippets (Id, body, title (figure 15)).
2) Le système supprime par la suite les mots vides, les mots latins et les caractères spéciaux.
3) Après, chaque snippet est utilisé pour générer l'arbre de suffixe «*Suffix Tree Document model* » (STDM) qui est utilisé pour trouver tous les clusters.
4) Un score est calculé pour chaque cluster afin d'obtenir le « *Base Clusters Graph* ».
5) Finalement, nous trouvons la racine de chaque étiquette en utilisant l'algorithme de khoja dans le but de fusionner tous les clusters similaires et éliminer ceux qui n'ont pas de signification.

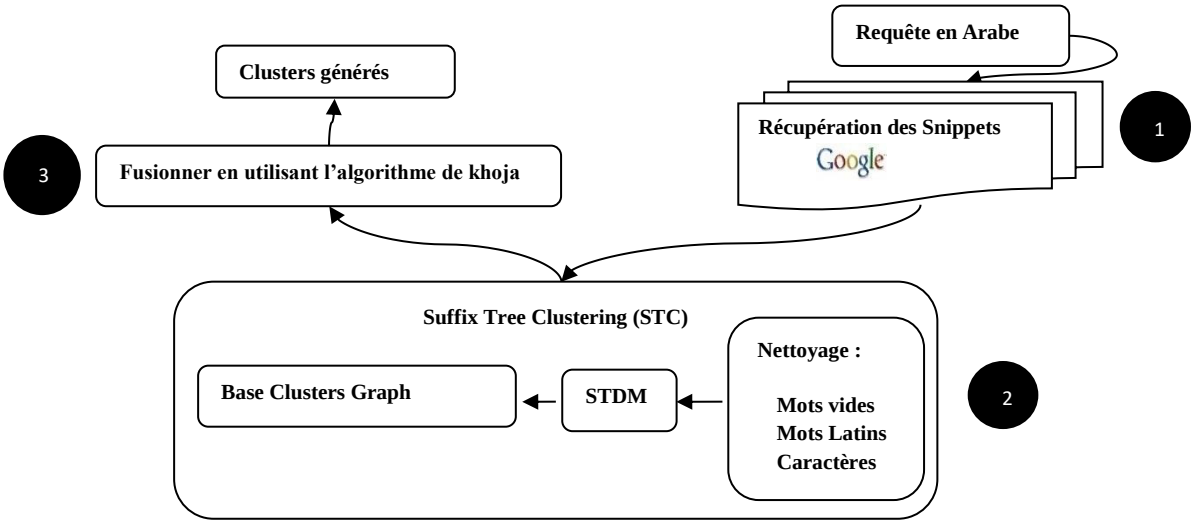


Figure 14 : *Système de Regroupent Thématique pour l'arbre des suffixes* [42]

```
<snippet>
  <id>0</id>
  <url>http://ar.wikipedia.org/wiki/%D8%AA%D8%B9%D9%84%D9%8A%D9%85</url>
  <body>التعليم والتربية هو بناء الفرد ونحو الأمية في المجتمع، وهو المحرك الأساسي في تطور  الحضارات ومحور قياس تطور ونماء المجتمعات فتقيم تلك المجتمعات على حسب نسبة . . .</body>
  <title>تعليم - ويكيبيديا، الموسوعة الحرة</title>
</snippet>
<snippet>
  <id>1</id>
  <url>http://www.mohe.gov.sa/</url>
  <body>يشمل الموقع كل ما يحتاجة المتصفح من معلومات خاصة  بالتعليم العالي بالمملكة  - المملكة العربية السعودية .</body>
  <title>وزارة التعليم العالي</title>
</snippet>
```

Figure 15 : *Exemple d'un Snippet en langue arabe*

2.1 Arbre des suffixes

Un arbre de suffixes « Suffix tree » est une structure de données qui permet de traiter les chaînes de caractères. Cette structure a été largement utilisée et a été appliquée à des problèmes fondamentaux des chaînes textuelles telles que la recherche de la plus longue chaîne répétée, la comparaison des chaînes de textuelles, et la compression de texte. L'arbre des suffixes traite couramment les chaînes comme des séquences de caractères, ou les documents comme des séquences de mots [43].

Un arbre de suffixes d'une chaîne est simplement un arbre compact de tous les suffixes de cette chaîne. L'arbre des suffixes a été utilisé d'abord par *Zamir et al.* [14] comme un algorithme de Clustering nommé *« Suffix Tree Clustering (STC) »*. C'est un algorithme de Clustering à temps linéaire qui est basé sur l'identification des phrases communes qui sont similaires dans des snippets afin de les regrouper en un seul cluster. Le STC a trois étapes logiques :

- Le *«Nettoyage»* du document
- Modèle de l'arbre des suffixes d'un document «*Suffix Tree Document model* »
- Identification des phrases communes « *Base Clusters Graph* »

2.2 «Nettoyage» du document

Dans cette étape, pour chaque snippet nous supprimons les mots vides comme (فانه, وان, والذي, وهذا, فكان, ستكون ...). Les mots vides[14] ont une fréquence élevée et une faible discrimination et doivent être filtrés. Ce sont des mots fonctionnels, généraux et communs de la langue qui ne contribuent généralement pas à la sémantique des documents et n'ont pas une valeur ajoutée. Généralement, la liste des mots vides consiste surtout d'une certaine combinaison de base de lettres et de numéros aussi bien que de pronoms, d'adverbes, de prépositions, quelques verbes, adjectifs et des conjonctions. Beaucoup de systèmes de recherche d'information tentent d'exclure ces mots vides de la liste de caractéristiques pour réduire l'espace de représentation et accroître leurs performances. En outre, dans notre cas, pour faire face notamment aux documents arabes, nous proposons également dans cette étape de supprimer les mots latins et les caractères spéciaux tels que (***$***, ***#*** ,..).

2.3 Modèle de l'Arbre des Suffixes d'un Document (STDM)

STDM est une structure de données arborescente contenant tous les suffixes d'un texte. Un suffixe dans notre cas est une partie d'une phrase. La phrase a une signification sémantique spécifique (les mots de la phrase sont ordonnés). Il considère une phrase $d = w_1, w_2, \ldots w_m$ comme une chaîne constituée de mots w_i $(i = 1, 2, \ldots, m)$. Une définition révisée de l'arbre des suffixes peut être présentée comme suit : Un arbre de suffixes généralisé pour un ensemble S de n chaînes, chacune d'une longueur m_k $(k = 1, 2, \ldots, n)$, est un arbre dirigé et enraciné avec exactement Σm_k feuilles marquées par un index de deux nombres (k, l) où l est compris entre 1 et m_k.

Chaque nœud interne, autre que la racine, a au moins deux enfants et chaque arête est étiquetée avec une sous chaîne non vide de mots d'une chaîne en S. Il n'y a pas deux arêtes hors d'un nœud pouvant avoir des étiquettes qui commencent par le même mot. Pour toute feuille (i, j), la concaténation des étiquettes de l'arête sur le chemin de la racine à la feuille (i, j) définit exactement le suffixe de S_i qui commence à la position j. La figure 16 montre un exemple d'arbre des suffixes généré d'un ensemble de trois chaînes arabes ou trois documents :

- Document 1 : "القط يأكل الجبن",
- Document 2 : "الفأريأكل الجبن أيضا"
- Document 3 : "القط يأكل الفأرأيضا".

[14] La liste des mots vides est téléchargée de https://github.com/mohataher/arabic-stop-words/blob/master/list.txt

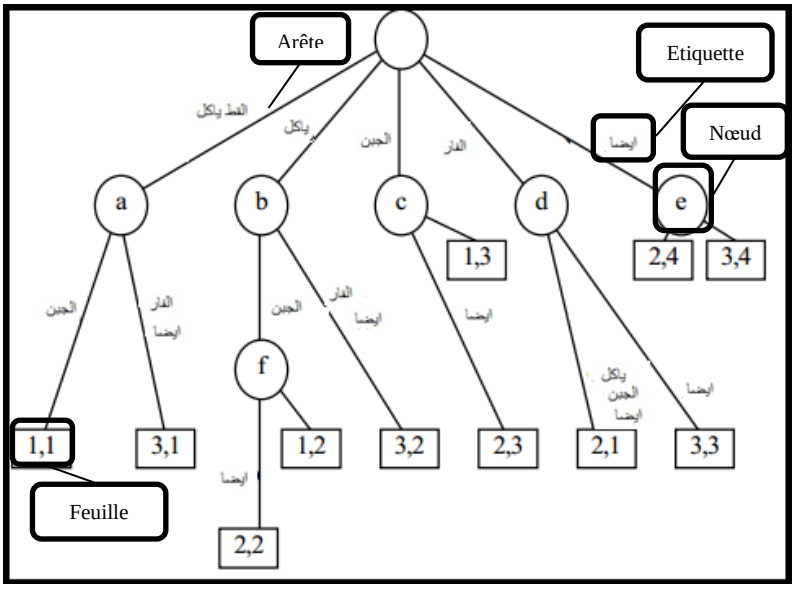


Figure 16 : *Modèle de l'Arbre des Suffixes de trois documents* [42]

Un arbre des suffixes des documents est un arbre compact contenant tous les suffixes de ces documents. Cet arbre est présenté par un ensemble de nœuds, de feuilles et d'étiquettes. L'étiquette d'un nœud de l'arbre est définie comme étant la concaténation, dans l'ordre, des sous-chaînes d'étiquetage des bords du chemin de la racine à ce nœud. Chaque nœud doit avoir un score et il est classé en fonction de son score. Le score est lié à :

- La longueur de l'étiquette des mots de nœud ;
- Le nombre d'occurrences de chaque mot (*Term Frequency*).

Chaque nœud de l'arbre des suffixes est marqué comme suit :

$$S(B)=|B|\times F(|P|) \times \sum_i^n TF.IDF(w_i) \quad \textbf{(3.1)}$$

$$F(|P|) = \begin{cases} |P|, & si\ 6 \geq |P| \geq 1 \\ 0, & si\ non \end{cases} \quad \textbf{(3.2)}$$

- Le cluster B étiqueté par le syntagme P
- |B| est le nombre de documents dans le cluster B.
- |P| est le nombre de mots qui construit le syntagme P.
- P=$\{w_1 w_2 \ldots . w_n\}$

Après, tous les clusters sont rangés selon leurs scores, et les meilleurs k-clusters sont sélectionnés pour le processus de fusionnement dans l'étape 3.

2.4 Etape de fusionnement pour générer la Base Clusters Graph

Le fusionnement est un processus où deux nœuds seront fusionnés selon la mesure de similarité. On définit la mesure de similarité binaire entre deux clusters basée sur le nombre des documents partagés. Pour deux Clusters B1 et B2, nous définissons la similarité **Similarity (B1, B2)** comme suit:

$$Similarity\ (B1, B2\ ) = \begin{cases} 1, si\ \left(\frac{|B1 \cap B2|}{|B1|}\right) \geq \alpha\ et \left(\frac{|B1 \cap B2|}{|B2|}\right) \geq \alpha \\ 0, si\ non \end{cases} \quad \textbf{(3.3)}$$

- α est une constante entre 0 et 1

Dans notre système, suite à plusieurs expériences nous proposons de fixer la valeur de α à 0.6. Deux nœuds avec une similarité égale à 1 sont connectés et le Base Cluster Graph est construit. Tous les clusters connectés sont considérés comme un seul cluster, qui contient l'union de tous les documents (Figure 17).

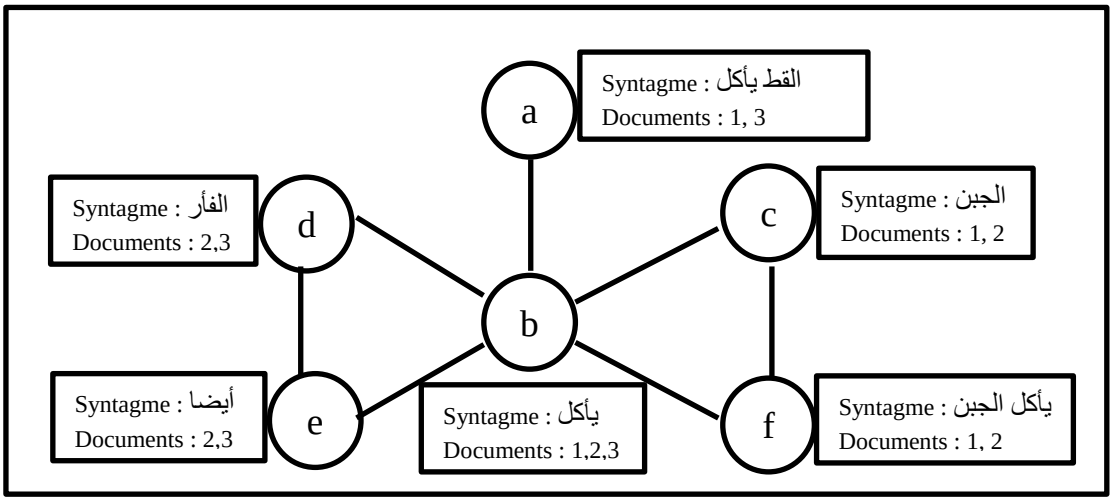


Figure 17 : *Base Cluster Graph* [42]

2.5 Adaptation de l'algorithme STC pour la langue arabe : Problème de prétraitement

La génération d'arbre de suffixe est basée sur la comparaison de chaque terme situé dans les snippets avec les autres termes. En effet, deux mots qui s'écrivent différemment alors qu'ils ont la même signification, ou deux mots un au singulier et l'autre au pluriel, sont considérés comme deux mots différents, et génèrent deux nœuds différents ce qui influence négativement le processus de fusionnement. Pour remédier à ce problème en langue anglaise, Zamir et al. [14] ont appliqué une étape de racinisation en utilisant l'algorithme de Porter [44] pour extraire la racine de chaque mot, et l'utiliser pour la comparaison dans le processus de construction d'arbre de suffixe. Pour le cas de langue arabe cette solution n'est pas efficace. Comme nous l'avons vu précédemment, nous avons montré que la forme graphique des mots ne peut être utilisée pour la constitution de familles morphologiques, en raison de l'agglutination et de la richesse dérivationnelle en langue arabe. L'exemple de figure 18 montre que l'utilisation du processus de prétraitement avant la génération de l'arbre de suffixe influence négativement la qualité des clusters en résultat, car nous avons obtenu plusieurs clusters n'ayant pas de relation avec la requête de l'utilisateur (Islam, الإسلام). Pour cela, nous proposons d'intégrer l'algorithme de racinisation Khoja [45] après la génération des clusters. En effet, la

génération du modèle de l'arbre des suffixes avant le processus de prétraitement permet de mettre en valeur les termes liés à la requête de l'utilisateur.

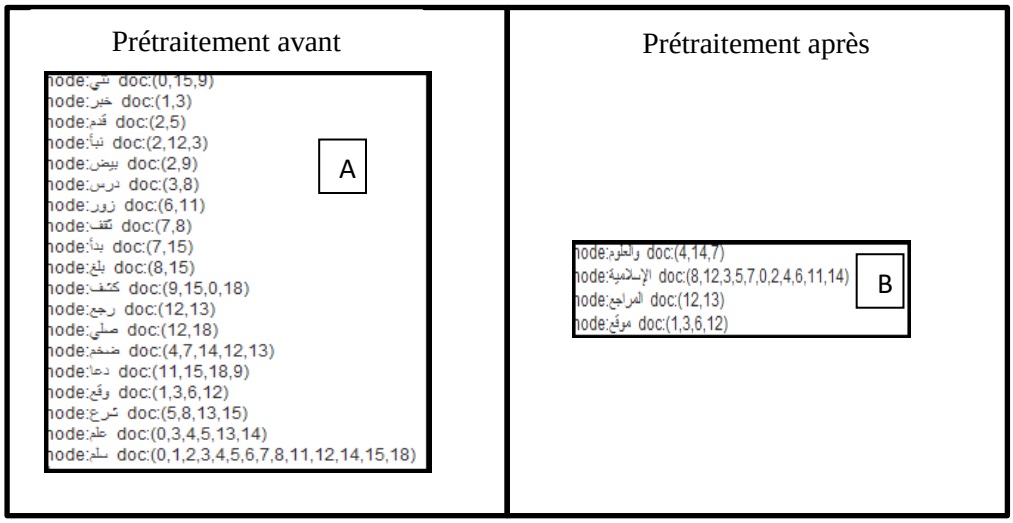


Figure 18 : *Comparaison entre prétraitement avant(A) et prétraitement après(B)*

2.6 AWSRC : plateforme proposée pour la visualisation thématique de résultats de recherche

La plateforme AWSRC que nous proposons pour visualiser les résultats de recherche organisés sous thème, se compose de cinq modules :

Module de Requête : une interface web permettant à l'utilisateur de saisir sa requête (Figure 19). La requête est envoyée au moteur de recherche google en utilisant son API. Les résultats retournés sous forme des snippets sont stockés dans un document XML.

Figure 19 : *Interface web de la plateforme AWSRC*

Module de Prétraitement : pour chaque snippet les mots vides, les mots latins et les caractères spéciaux sont supprimés.

Module de Clustering : utilise l'algorithme STC pour générer les clusters.

L'utilisation de l'algorithme de Khoja : l'intégration de l'algorithme de racinisation Khoja, après la génération des clusters pour le fusionnement des clusters similaires.

Module d'affichage des Résultats : permet d'afficher une interface web, qui se compose de deux parties : 1) le côté droit présente les labels des clusters et le nombre des documents dans chaque cluster, 2) le côté gauche affiche le contenu de chaque cluster sélectionné (Figure 20).

Figure 20 : *Interface web de l'affichage des résultats de requête (التعليم العالي)*

2.7 Système de recherche d'information interactif

Généralement, les résultats d'un système de clustering de résultats de recherche ne sont pas adaptés avec l'interaction de l'utilisateur. En effet, les utilisateurs modifient fréquemment la requête de recherche précédente afin de récupérer de meilleurs résultats, du fait qu'ils ne peuvent pas exprimer leurs besoins exacts. Ceci conduit à l'utilisation de mots-clés ou des requêtes inappropriées. Ce cycle est répété jusqu'à ce que l'utilisateur est satisfait ou abandonne sa recherche. Ce qui conduit à consommer beaucoup de temps afin que l'utilisateur satisfait son besoin.

Pour résoudre ce problème, nous proposons un nouveau système interactif pour les utilisateurs du "web arabe" [46], qui peuvent décider d'un coup d'œil si le contenu d'un cluster est intéressant. L'utilisateur n'a pas besoin de reformuler sa requête, il peut simplement cliquer sur le label qui décrit plus précisément son besoin dans la hiérarchie des sujets. Par conséquent, l'utilisateur peut naviguer dans le sens de généralisation ou de spécification.

La figure 21 présente notre proposition pour rendre le système interactif. Cette approche peut être résumée en trois étapes comme suit :

1) L'utilisateur spécifie sa requête en utilisant l'interface Web, cette requête est envoyée à l'AWSCR pour générer les Clusters.
2) L'utilisateur clique sur le label de cluster qui lui apparait intéressant, et le système affiche à l'utilisateur les snippets associés à ce label.
3) Simultanément avec l'étape 2, le label choisi est envoyé en tant qu'une nouvelle requête au système, afin de régénérer à l'utilisateur d'autres labels qui sont des sous-domaines du premier.

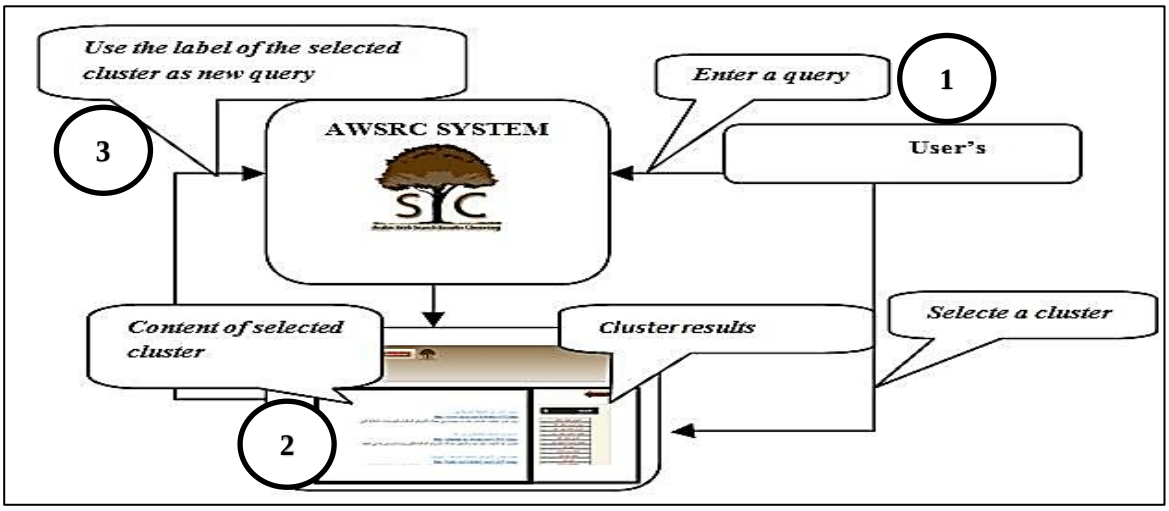


Figure 21 : *Système de recherche d'information interactif* [46]

La figure 22 présente un exemple qui illustre le fonctionnement de notre système interactif. Quand l'utilisateur clique sur le cluster (الاسواق, Markets), le système commence par afficher tous les résultats relatifs à cette requête.

Par la suite, lorsque l'utilisateur choisit d'afficher le cluster en cliquant sur (الاسواق المالية, les bourses), le système affiche les résultats correspondants, et présente aussi d'autres sous-clusters du sous-domaine de ce dernier tels que (سوق الأوراق المالية, bourse) et (الاسواق المالية العربية, les bourses arabes). Le même principe est appliqué pour le cluster (الاسواق المالية العربية, les bourses arabes).

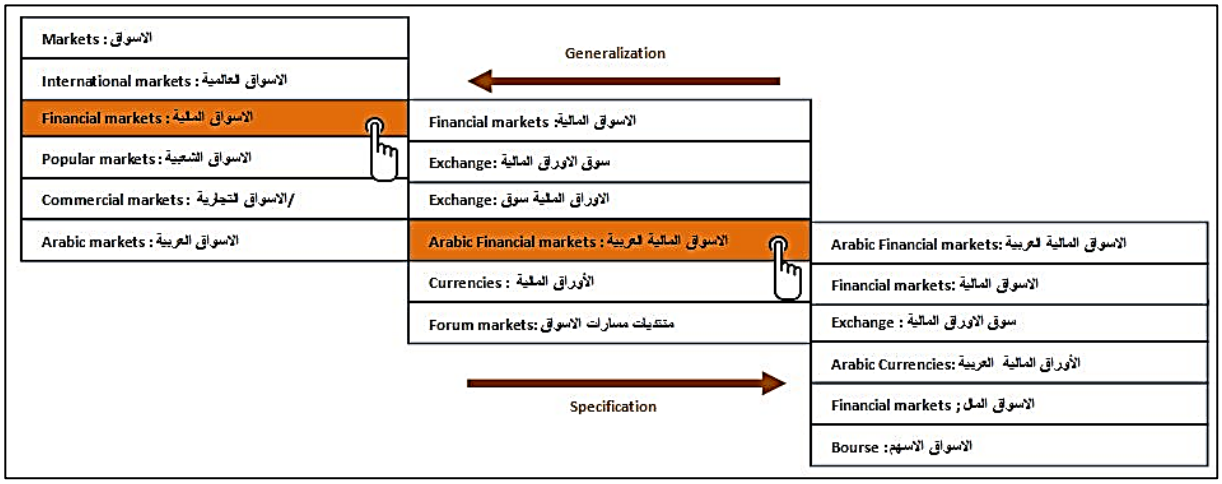


Figure 22 : *Exemple d'utilisation du Système interactif* [46]

3 Résultats Expérimentaux

Généralement l'évaluation des techniques classiques de regroupement peut être divisée en deux approches : l'approche objective et l'approche subjective. Dans l'approche objective, la partition originale est connue et est comparée aux clusters générés. La comparaison peut être effectuée en utilisant des mesures traditionnelles du domaine d'évaluation du clustering telles que l'entropie et la pureté [47].

Cependant, l'évaluation objective des algorithmes de recherche SRC est très difficile. En effet, il n'existe pas une collection standard de tests de la langue arabe pour évaluer les algorithmes SRC, car nous n'avons pas des connaissances préalables sur les clusters initiaux correspondants aux requêtes des utilisateurs. Par conséquent, nous proposons d'utiliser uniquement l'approche subjective. Afin d'évaluer et d'illustrer l'efficacité de notre système proposé, nous présentons et discutons les meilleurs résultats de quelques requêtes arabes. Les résultats obtenus sont également comparés avec les trois autres systèmes existants de regroupement de résultats de recherche que sont **Clusty**[15], **IBoogie**[16], **Yippy**[17].

Requête	**التعليم**	**السيارات**	**السياحة**
Résultats	التعليم	اخبار السيارات	السياحة
	وزير التعليم	العاب سيارات	السياحة نشاط
	التربية والتعليم	سباقات السيارات	السفر
	مركز	اخر اخبار	الخدمات
	بناء	العاب سباق	مصر

Tableau 13 : *Exemples des résultats des recherches de AWSRC* [46]

Ce tableau présente les cinq meilleurs résultats de notre système appliqué sur quelques requêtes arabes. Nous constatons que la plupart de ces étiquettes de clusters de chaque requête sont des mots-clés appartenant au même domaine. Dans la section suivante, nous utilisons une autre requête «السياحة» en utilisant les trois fameux web post-récupération Système Clusty (Figure 23), IBoogie (Figure 25) et Yippy (Figure 24) (anciennement Clusty), afin de comparer leurs résultats avec les résultats obtenus en utilisant notre système proposé (AWSRC) (Figure 26). Nous pouvons conclure des figures 23, 24, 25 et 26 ce qui suit (tableau 14).

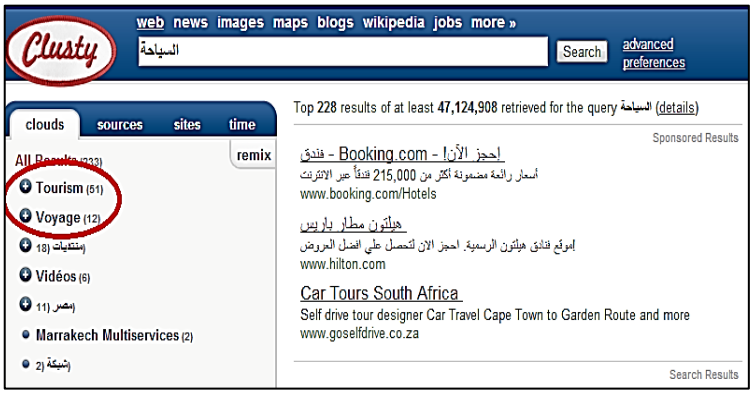


Figure 23 : *Clusty (requête السياحة)*

[15] http://clusty.com/
[16] http://iboogie.com/
[17] http://www. yippy.com /

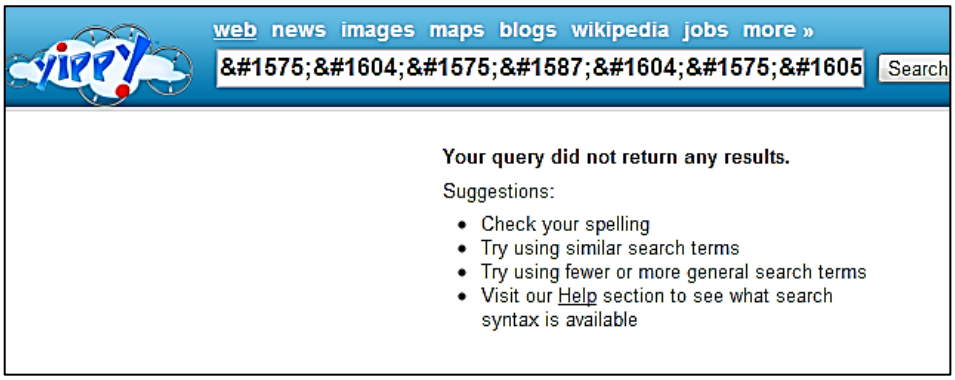


Figure 24 : *Yippy(requête السياحة)*

Figure 25 : *IBoogie(requête السياحة)*

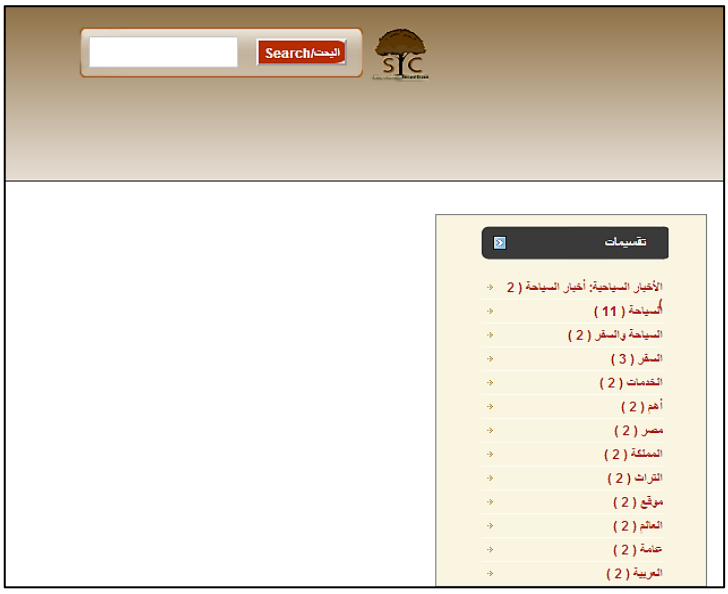


Figure 26 : *AWSRC (requête السياحة)*

Dans le tableau 14, nous nous basons sur deux facteurs que nous estimons déterminants pour évaluer le système avec la langue arabe :

1. L'affichage uniquement des résultats en arabe.
2. Comment la langue arabe a-t-elle été traitée.

	Clusty	**IBoogie**	**Yippy**	**AWSRS**
Comment le système traite la langue arabe?	Utilise « *Cross langage* »	Utilise « *Cross langage* »	Pas de langue arabe	Langue arabe
L'affichage des résultats en langue arabe	Mélange de langage pour résultats	Résultats en langue latine	Pas des résultats en arabe	Résultats en arabe

Tableau 14 : *Etude comparative*

A partir du tableau 14, notre système est efficace et est plus adapté pour la langue arabe que les autres systèmes existants de regroupement de résultats de recherche.

4 Conclusion

La consultation des résultats de recherche est l'un des principaux problèmes avec les moteurs de recherche Web traditionnels (Google, Yahoo et Bing) pour l'anglais, et les autres langues en général et pour l'arabe en particulier. L'organisation des résultats de recherche en arabe sous forme de clusters facilite aux utilisateurs arabes l'accès rapide à ces résultats.

L'algorithme Suffix Tree Clustering (STC) a été largement utilisé avec différentes versions adaptées pour l'anglais, le chinois et d'autres langues, mais n'a jamais été appliqué pour la langue arabe.

Dans ce chapitre, nous avons montré que l'algorithme STC ne peut pas être appliqué directement à la langue arabe. Nous avons proposé d'implémenter un nouveau schéma qui intègre correctement l'algorithme STC dans notre système de regroupement de résultats de recherche arabe.

L'approche proposée regroupe automatiquement les résultats de recherche avec des labels compréhensifs ayant un sens. Des expériences et des évaluations préliminaires sont menées et les résultats expérimentaux montrent que notre proposition est efficace et facilite aux utilisateurs arabes la consultation rapide à travers les résultats de la recherche.

CHAPITRE 4. Analyse de concepts formels pour le regroupement thématique des résultats de SRI

1 Introduction

Dans ce chapitre, nous proposons une deuxième contribution que nous avons apportée sur ce sujet de consultation. Cette contribution se base principalement sur l'utilisation de la FCA comme un formalisme mathématique [48]. FCA a été utilisée avec succès comme une nouvelle méthode pour regrouper les résultats de recherche qui se base sur le regroupement conceptuel. Elle a été intégrée dans plusieurs systèmes afin de remédier au problème de consultation sur le Web, spécialement pour les langues latines [49], [26] . Cependant, FCA n'a jamais été utilisée pour le WSRC arabe afin de remédier au problème de la consultation. En outre, la langue arabe a ses propres propriétés qui sont très différentes des langues latines. De ce fait, l'utilisation directe pour la langue arabe de n'importe quel système de clustring de résultats de recherche dédiés aux autres langues peut avoir un impact négatif sur les résultats de la représentation thématique à base des clusters.

2 Analyse Formelle de Concepts

FCA est une théorie qui constitue un pont entre les mathématiques et la représentation de connaissances [48]. C'est la restructuration de la théorie des treillis en adéquation avec la philosophie de la pensée humaine. Elle vise à identifier des clusters des connaissances, appelés concepts formels, et à ordonner ces clusters sous la forme de treillis. Dans cette contribution, l'idée de base de l'utilisation du modèle FCA est d'explorer le contexte formel entre les snippets retournés par les moteurs de recherche, et de construire le treillis « *concept lattice* » (Figure 27).

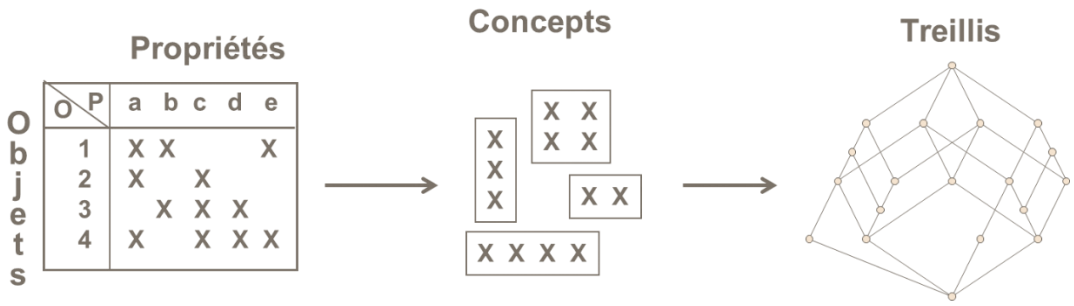


Figure 27 : *Modèle de l'analyse formelle de concepts*

2.1 Contexte Formel (G, M, I)

L'Analyse Formelle de Concepts se base sur la notion de concept formel qui est considérée du point de vue psychologique comme l'unité de base de la pensée humaine. Un concept peut être défini comme un ensemble d'objets et de leurs propriétés communes. Ces concepts formels sont extraits à partir d'une relation binaire définie entre un ensemble d'objets et un ensemble de propriétés (attributs). Cette relation est appelée ***Contexte Formel*** et peut être représentée sous la forme d'un tableau où les lignes correspondent aux objets et les colonnes correspondent aux propriétés. Du point de vue mathématique, un ***Contexte Formel*** est un triplet (G, M, I) où G et M sont des ensembles et I ⊆ G ×

M. Les éléments de G sont appelés les objets et ceux de M sont appelés les attributs. L'ensemble de couple I est considéré comme une relation et est donc noté g I m au lieu de (g, m) ∈ I ce qui se dit : « l'objet g possède l'attribut m». Le tableau 15 présente un exemple de Contexte Formel :

- Les Objets sont représentés par l'ensemble des documents (G1, G2, .., G9).
- Les Attributs sont représentés par l'ensemble des terms Arabes.
- (0 ,1) représente la relation binaire entre les termes et les objets.

	جغوار	السيارة	المركبة	نموذج	الرياضة	حيوان	فهد
G1	1	1	0	0	0	0	0
G2	1	1	0	1	0	0	0
G3	0	1	0	0	1	0	0
G4	1	0	0	0	0	1	0
G5	1	0	1	0	0	0	0
G6	1	1	0	0	0	0	0
G7	0	0	1	1	0	0	0
G8	1	0	0	0	0	1	1
G9	1	1	0	0	1	0	0

Tableau 15 : *Exemple de Contexte Formel*

2.2 Concept Formel du Contexte Formel (G, M, I)

Un Concept Formel du contexte Formel (G, M, I) [48] est une paire (A, B) où A ⊆ G et B ⊆ M qui vérifie $A=B^I$ et $B=A^I$ où :

$$A^I = \{m \in M \mid gIm \ \forall g \in A\} \quad \textbf{(4.1)}$$

$$B^I = \{g \in G \mid gIm \ \ \forall m \in B\} \quad \textbf{(4.2)}$$

- A^I est l'ensemble des attributs partagés par tous les objets de A.
- B^I est l'ensemble des objets qui possèdent tous les attributs de B.
- On dit que A est l'extension du concept formel et B est son intention.

L'ensemble de tous les concepts formels (noté β(G,M,I)) muni de la relation d'ordre **" ≤ "** forme un treillis complet dont la structure est donnée par le théorème fondamental de Ganter et Wille [48].

2.3 Treillis β(G,M,I)

L'ensemble de tous les concepts formels présente une propriété algébrique importante : il constitue un treillis complet « Concept Lattice ». Le treillis associé au contexte formel de l'exemple précédent (tableau 15) est représenté dans la figure 28. Du point de vue mathématique, le treillis β(G,M,I) est une hiérarchie ordonnée de tous les concepts formels du contexte formel (G, M, I).

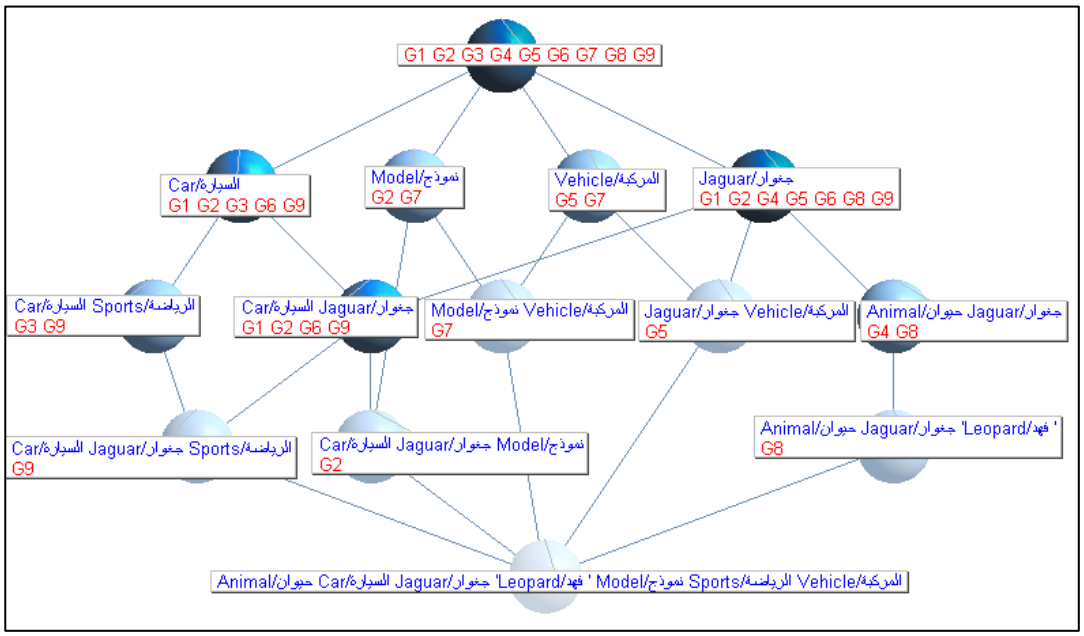


Figure 28 : *Exemple de concept lattice généré par le contexte formel du Tableau 15*

3 Algorithmes de construction de treillis de concepts

Pour construire un treillis de concepts formels, il faut d'abord énumérer les rectangles maximaux (les fermés), ensuite trouver les relations d'ordre partiel entre ces rectangles, et enfin représenter graphiquement le treillis (construction du diagramme de HASSE). En considérant la stratégie d'acquisition de données à partir d'un contexte formel, trois familles d'algorithmes sont mises en évidence :

- Les algorithmes **batch** qui considèrent la totalité du contexte dès le départ,
- Les algorithmes **incrémentaux** qui considèrent le contexte ligne par ligne,
- Les algorithmes d'**assemblage** qui répartissent le contexte formel puis calculent les concepts formels correspondants à chaque partie ensuite ils font l'assemblage.

3.1 Les algorithmes batch

Ils présentent les algorithmes de la première génération. L'entrée de ces algorithmes est le contexte formel tout entier. Ils calculent les concepts formels et l'ordre entre ces concepts formels simultanément ou séquentiellement. Nous trouvons dans cette catégorie l'algorithme de Chein [50], qui est l'un des premiers algorithmes proposés. C'est un algorithme itératif qui repose sur la propriété suivante :

Soit K= (G, M, I) un contexte formel et (A_1, B_1) et (A_2, B_2) deux paires telles que :

- $A_1, A_2 \subseteq G$ et $B_1, B_2 \subseteq M$
- $A_1^I = B_1$ et $A_2^I = B_2$

Le rectangle ($A_1 \cup A_2$, $B_1 \cap B_2$) est tel que :

- $(A_1 \cup A_2)^I = (B_1 \cap B_2)$
- $(B_1 \cap B_2)^I = (A_1 \cup A_2)$

Est un élément de β (G, M, I) si seulement s'il est maximal.

L'algorithme construit le treillis des concepts formels comme suit :

- Initialement, l'algorithme part d'un ensemble L_1 de paires (A, B) représentant les lignes du contexte formel (A contient un seul élément de G et $B = A^I$).
- Un élément (A_3, B_3) de L_{i+1} est obtenu en combinant deux éléments (A_1,B_1) et (A_2,B_2) de L_i comme suit: $A_3 = (A_1 \cup A_2)$ et $B_3 = (B_1 \cap B_2)$ tel que $(A_1 \cup A_2)^I = (B_1 \cap B_2)$, $(B_1 \cap B_2)^I = (A_1 \cup A_2)$
- Les éléments de L_i inclues dans au moins un élément de L_{i-1} ne sont pas maximaux et sont donc supprimés.
- L'algorithme s'arrête lorsque L_{i+1} contient moins de deux éléments. Les éléments non supprimés après l'arrêt de l'algorithme sont les concepts formels du contexte formel considéré.

Le tableau 17 présente un exemple de déroulement de l'algorithme de **Chein** en utilisant le contexte formel du Tableau16 [51] :

	a	**b**	**c**	**d**	**e**
1	1	1	0	0	0
2	1	0	1	0	0
3	0	1	1	1	0
4	0	0	0	1	1
5	0	0	0	1	0
6	1	1	1	1	1

Tableau 16 : *Contexte formel d'entrée*

L_1	L_2	L_3
~~({1},{a,b})~~ ~~({2},{a,c})~~ ~~({3},{b,c,d})~~ ~~({4},{d,e})~~ ~~({5},{d})~~ ({6},{a,b,c,d,e})	~~({1,2},{a})~~ ~~({1,3},{b})~~ ({1,6},{a,b}) ~~({2,3},{c})~~ ({2,6},{a,c}) ~~({3,4,5},{d})~~ ({3,6},{b,c,d}) ({4,6},{d,e})	({1,2,6},{a}) ({1,3,6},{b}) ({2,3,6},{c}) ({3,4,5,6},{d})

Tableau 17 : *Trace de l'algorithme de Chein*

Un autre algorithme dans cette catégorie est **NextClosure** [52]. Il s'appuie sur l'ordre lexicographique entre ensembles d'attributs pour calculer les fermés. Le calcul des fermés peut être appliqué aux attributs (NextIntent) auquel cas on obtient les intentions des concepts formels, ou bien appliqué aux objets (NextExtent) et dans ce cas on obtient les extensions des concepts formels.

L'algorithme **Bordat** [53] construit les concepts formels en s'appuyant sur une structure d'arbre pour garder les résultats intermédiaires. L'algorithme **Close-by-One** [54] utilise une technique similaire à **NextClosure** pour la génération des concepts formels et une structure particulière appelée *arbre CbO* pour garder en mémoire les concepts formels générés. L'algorithme **Titanic** [55] utilise la

notion de fréquence pour calculer les concepts formels sans faire d'intersection entre les ensembles d'attributs.

3.2 Algorithmes incrémentaux

Les algorithmes incrémentaux remédient au problème de la reconstruction du treillis dans le cadre de contextes dynamiques. Ces algorithmes effectuent des mises à jour locales du treillis après l'ajout d'un objet dans le contexte formel.

A la différence des algorithmes batch, les algorithmes incrémentaux considèrent le contexte formel ligne par ligne (colonne par colonne) et construisent le treillis de concepts formels par ajouts successif de ligne (colonne) tout en conservant sa structure.

Dans cette catégorie figure l'algorithme de **Norris** [56] où on considère le tableau ligne à ligne :

- Initialement L_1 ne contient qu'une seule ligne (g_1,B_1) où $g_1^I=B_1$.
- Une étape dans cet algorithme consiste à construire L_k connaissant g_k et L_{k-1}.
- Une paire (A_i,B_i) est soit étendue en $(A_i \cup g_k,B_i)$ si B_i est inclus dans $b_k= g_k^I$ soit recopié et on ajoute à L_k le rectangle $(A_i \cup g_k,B_i \cap b_k)$ si $B_i \cap b_k$ n'est pas une intention du rectangle L_k.

Considérant l'exemple de contexte formel présenté dans le tableau 16, le déroulement de l'algorithme de **Norris** est comme suit (tableau 18) :

L_1	L_2	L_3	L_4	L_5	L_6
({1},{a,b})	({1,2},{a}) ({1,6},{a,b})	({1,2},{a}) ({1,3},{b}) ({1,6},{a,b}) ({2,3},{c}) ({2,6},{a,c}) ({3,6},{b,c,d})	({1,2},{a}) ({1,3},{b}) ({1,6},{a,b}) ({2,3},{c}) ({2,6},{a,c}) ({3,4,5},{d}) ({3,6},{b,c,d}) ({4,6},{d,e})	({1,2},{a}) ({1,3},{b}) ({1,6},{a,b}) ({2,3},{c}) ({2,6},{a,c}) ({3,4,5},{d}) ({3,6},{b,c,d}) ({4,6},{d,e})	({1,2},{a}) ({1,3},{b}) ({1,6},{a,b}) ({2,3},{c}) ({2,6},{a,c}) ({3,4,5},{d}) ({3,6},{b,c,d}) ({4,6},{d,e}) ({6},{a,b,c,d,e})

Tableau 18 : *Trace de l'algorithme de Norris*

Dans cette même catégorie, on trouve l'algorithme **Galois** [57] et l'algorithme de **Godin** [58].

3.3 Algorithmes d'assemblage

Le seul algorithme connu de cette famille est l'algorithme **Divide&Conquer** [59] qui permet de diviser un contexte formel en deux parties verticalement ou horizontalement puis de calculer le treillis de concepts formels correspondant à chaque partie et enfin d'assembler les treillis obtenus en un seul. La stratégie d'assemblage de treillis est bien adaptée aux problèmes d'intégration de vues partielles sur un domaine.

Le tableau 20 présente un exemple de déroulement de l'algorithme de de **Divide&Conquer** en utilisant le contexte formel du tableau19 [51].

	a	b	c	d	e	f	g
1	1	0	0	1	0	0	1
2	1	0	0	1	0	0	1
3	1	0	0	1	0	1	0
4	1	0	0	1	0	1	0
5	0	0	1	0	1	1	0
6	0	0	1	0	1	1	0
7	0	1	0	0	1	1	0
8	0	1	0	0	1	1	0
9	1	0	0	0	1	1	0

Tableau 19 : *Contexte formel d'entrée*

On le divise en deux contextes formels, puis on construit le treillis qui correspond à chacun d'eux :

	a	b	c	d
1	1	0	0	1
2	1	0	0	1
3	1	0	0	1
4	1	0	0	1
5	0	0	1	0
6	0	0	1	0
7	0	1	0	0
8	0	1	0	0
9	1	0	0	0

	e	f	g
1	0	0	1
2	0	0	1
3	0	1	0
4	0	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	1	0

Tableau 20 : *Trace de l'algorithme de Divide&Conquer*

A la fin, on assemble les deux treillis dans un seul comme suit (Figure 29):

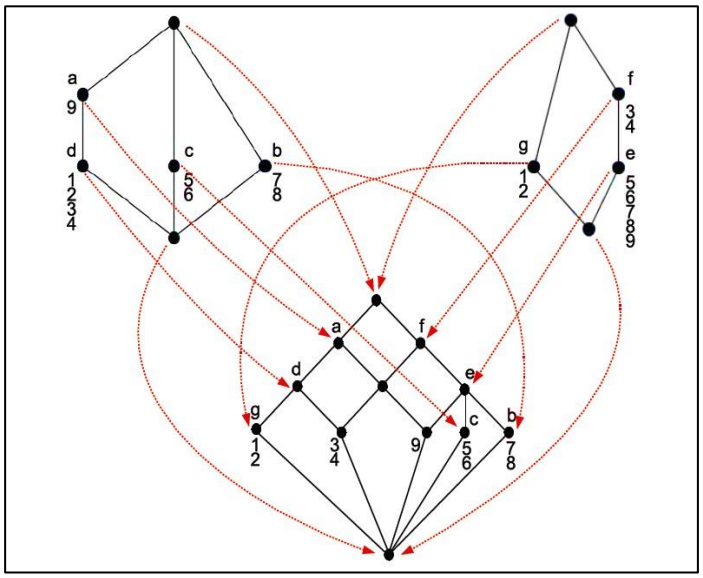


Figure 29 : *Treillis résultant du déroulement de l'algorithme de Divide&Conquer*

Le tableau 21 recense la complexité des algorithmes de construction de treillis :

Algorithme	Complexité en temps
Chein	$O(\|G\|^3\|M\|\| \beta \|)$
Bordat	$O(\|G\|\|M\|^2\| \beta \|)$
Ganter	$O(\|G\|^2\|M\|\| \beta \|)$
Norris	$O(\|G\|^2\|M\|\| \beta \|)$

Tableau 21 : *Complexité des algorithmes de construction de treillis* [60]

Le choix d'un algorithme de construction de treillis doit être basé sur les propriétés des données à traiter. Dans [60], les auteurs recommandent :

- L'utilisation de l'algorithme de **Godin** pour les contextes petits et clairsemés.
- L'utilisation de **Norris** pour les contextes denses.
- L'utilisation de l'algorithme de **Bordat** pour les contextes d'une densité moyenne et surtout quand il s'agit de construire le diagramme.

4 Outils pour la génération de Treillis

Dans cette section, nous passons en revue les outils qui ont été développés pour construire, manipuler et visualiser des treillis de concepts formels. Nous nous limitons ici aux outils couramment utilisés en recherche académique, une liste plus exhaustive peut être consultée sur FCA homepage[18].

4.1 ConImp

Proposé en 1986, ConImp [61] abréviation de ***Con****texts and* ***Imp****lications* est l'un des plus anciens outils de manipulation de treillis. Il est disponible sous DOS et Linux. Il fonctionne en mode texte et ne permet pas de visualiser les treillis.

- format entrée contexte : cxt
- format sortie contexte : cxt
- format entrée treillis : non
- format sortie treillis : bgr
- visualisation : non
- adresse : www.mathematik.tu-darmstadt.de/~burmeister/

4.2 Galicia

Galicia [59] présente certaines fonctionnalités avancées comme la manipulation de contextes relationnels, la fusion de treillis ou encore la construction d'icebergs de concepts formels.

- format entrée contexte : slf, bin.xml, ibm
- format sortie contexte : slf, bin.xml
- format entrée treillis : lat.xml
- format sortie treillis : lat.xml

[18] http://www.upriss.org.uk/fca/fca.html

- visualisation : oui (Figure 30)
- adresse : www.iro.umontreal.ca/~galicia/

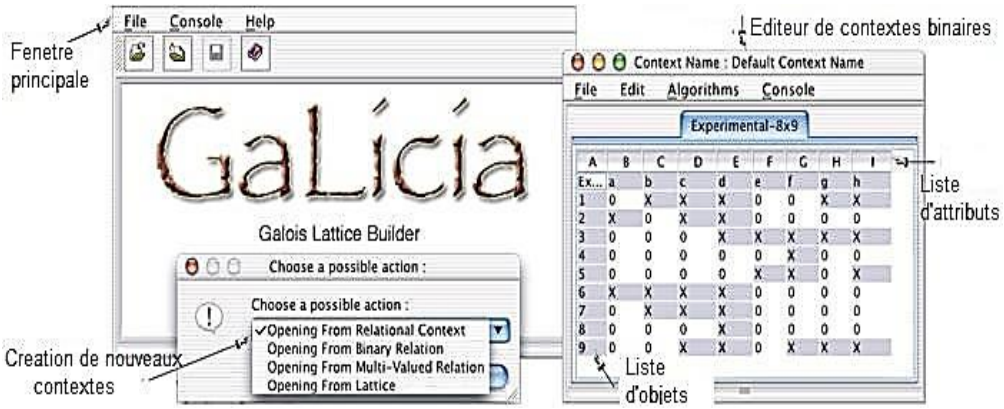


Figure 30 : *Interface de Galicia*

4.3 ConExp

ConExp (*Concept Explorer*) [62] a d'abord été développé dans le cadre d'une thèse de maîtrise sous la supervision du Professeur Tatyana Taran à l'Université technique nationale d'Ukraine «KPI» en 2000. Il supporte seulement les contextes binaires. Il est caractérisé par sa facilité de manipulation.

- format entrée contexte : cex, cxt, csv, oal
- format sortie contexte : cex, cxt
- format entrée treillis : cex
- format sortie treillis : cex
- visualisation : oui
- adresse : conexp.sourceforge.net/

4.4 Toscana

Parmi les logiciels les plus récents, on distingue Toscana [63]. Toscana est actuellement développé en Java (ToscanaJ) par des équipes des universités de Darmstadt en Allemagne et du Queensland d'Australie. L'une des particularités de ToscanaJ est de construire et visualiser des treillis entrelacés.

- format entrée contexte : csx, cxt, csc (format Anaconda), xml (format Cernato)
- format sortie contexte : csx
- format entrée treillis : csx
- format sortie treillis : csx
- visualisation : oui, y compris nested-line diagrams (Figure 31)
- adresse : toscanaj.sourceforge.net/

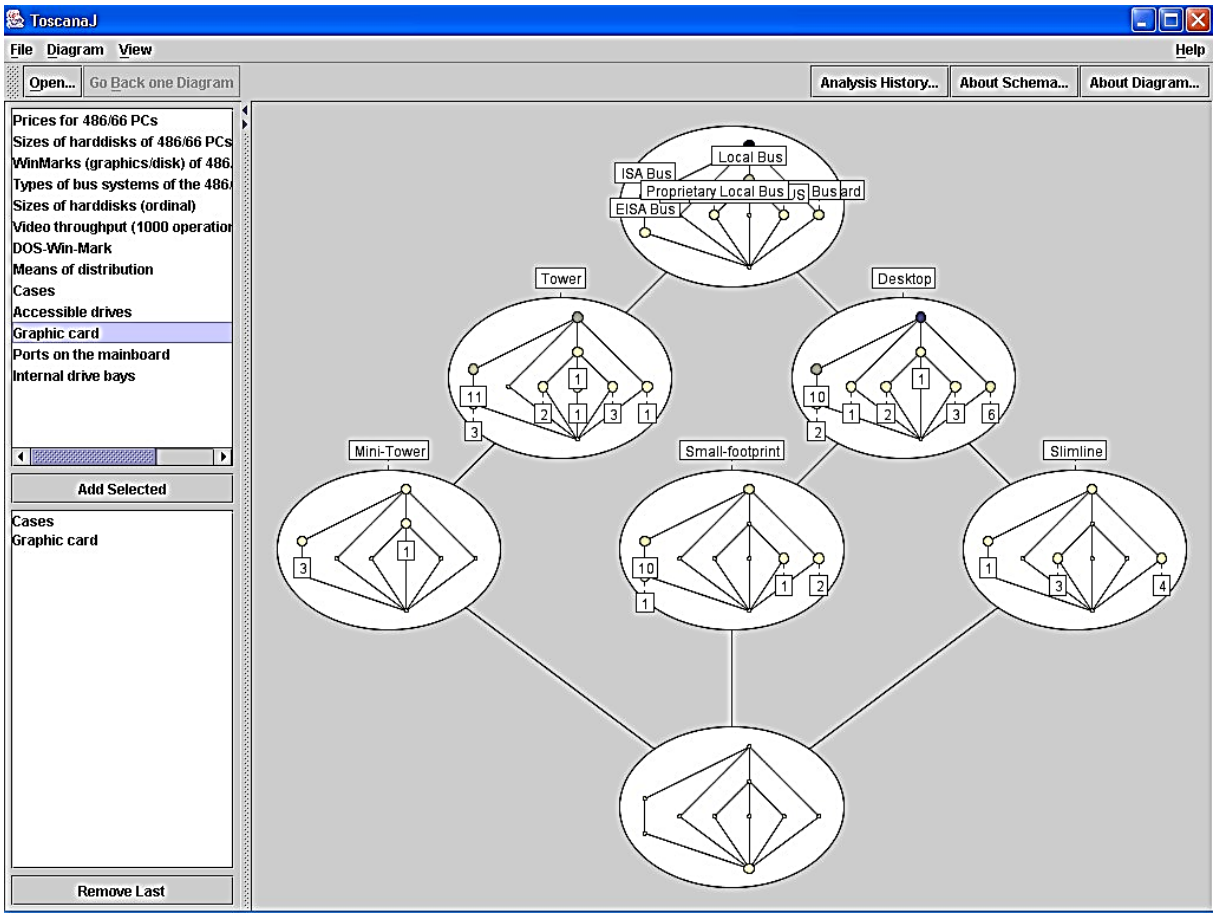


Figure 31 : *Interface ToscanaJ*

5 Analyse de concepts formels pour le Regroupement thématique des résultats de SRI

Pendant ces dernières années, de nouvelles approches de recherche d'information par treillis ont émergé. Il a été constaté que les SRI utilisant les treillis deviennent complexes quand il s'agit de grandes masses d'informations. C'est pour cette raison que ces nouvelles approches proposent d'utiliser FCA à un méta niveau, c'est-à-dire au-dessus d'un moteur de recherche classique.

Ces méta-moteurs servent d'intermédiaire entre l'utilisateur et les moteurs de recherche tels que Google et Yahoo. Leur objectif est de diminuer la charge informationnelle de l'utilisateur en améliorant la présentation des résultats de recherche de ces moteurs. Ces approches fonctionnent suivant le même principe : d'abord l'utilisateur formule sa requête qui sera transmise à un moteur de recherche externe. Ce dernier retourne les résultats de recherche, puis un treillis de concepts formels est construit à partir des mots du titre et de l'extrait « *Snippet* » des documents retournés. Ce type d'approche est implémenté dans plusieurs systèmes opérationnels. Dans ce qui suit, nous allons passer en revue ces systèmes, en présentant les fonctionnalités qu'ils offrent à l'utilisateur.

5.1 CREDO

Le système CREDO [26], acronyme de Conceptual REorganisation of DOcuments, ou son adaptation CREDINO pour les PDA [57], utilise la structure de treillis pour organiser l'ensemble des documents retournés par un moteur de recherche externe de façon à ce que l'utilisateur ait une vision globale de l'espace de recherche et pourra ainsi naviguer aisément parmi les pages retournées. Afin d'atteindre son objectif CREDO procède comme suit :

Interaction avec le moteur de recherche :

L'interaction entre CREDO et le moteur de recherche est gérée par le protocole SOAP *(Simple Object Access Protocol).* La requête doit d'abord être encodée puis émise vers le moteur de recherche. CREDO collecte les 100 documents retournés par le moteur de recherche. Chaque document retourné est composé d'un titre, d'un extrait et d'une URL (*Uniform Resource Locator*).

Indexation des documents retournés :

Un problème majeur de la construction de treillis des concepts formels des documents sélectionnés est la génération de certains concepts dépourvus de sens en raison de mauvaises combinaisons de termes dans les documents. Une façon de résoudre ces problèmes est de décrire un document par un ensemble limité de mots. CREDO considère alors seulement les informations contenues dans les résultats retournés par le moteur de recherche. Il se concentre sur les éléments qui décrivent le mieux le contenu des documents à savoir le titre et l'extrait. Chaque document est indexé alors par deux ensembles de termes, un pour le titre et un autre pour l'extrait.

Construction de la hiérarchie :

CREDO construit un premier niveau de la hiérarchie en exploitant le contexte *document* × *mots du titre.* Cependant, il est probable que de nombreux documents ne partagent aucun terme avec les autres documents, restants ainsi dissociés. CREDO utilise donc une approche hybride, c'est-à-dire, l'utilisation d'un contexte *document* × *(mots du titre+ de l'extrait)*. Ce dernier va servir à la construction des autres niveaux de la hiérarchie. Un algorithme spécifique est proposé pour construire une hiérarchie à partir de deux contextes. De toute évidence, la structure résultante n'est pas un vrai treillis de concepts formels, dans le sens où il ne peut pas être considéré comme le treillis de concepts formels d'un contexte spécifique.

Visualisation des résultats et interaction avec la hiérarchie :

Après la construction de la hiérarchie, CREDO présente les résultats à l'utilisateur via une interface illustrée dans la figure 32. Le panneau de gauche présente les intentions des concepts formels du premier niveau du treillis (les fils directs de $\top$). Lorsque l'utilisateur sélectionne une intention, les pages en extension du concept formel sont listées sur le panneau de droite et les intentions des fils directs du concept formel sélectionnés sont affichées sur le tableau de gauche. Le treillis est donc présenté sous forme d'arbre qui sert de support à l'exploration de l'ensemble des documents. L'accès aux documents se fait en sélectionnant des concepts formels de plus en plus spécifiques, auxquels sont

rattachés des documents. Un intérêt majeur du treillis est l'héritage multiple entre concepts formels, un même concept formel, et donc un même ensemble de documents, est atteignable par plusieurs chemins (i.e. un concept formel avec héritage multiple se retrouve dupliqué dans l'arbre). Comme CREDO permet à l'utilisateur de naviguer à travers les résultats de la requête, cela peut être considéré comme une forme de raffinement de requêtes.

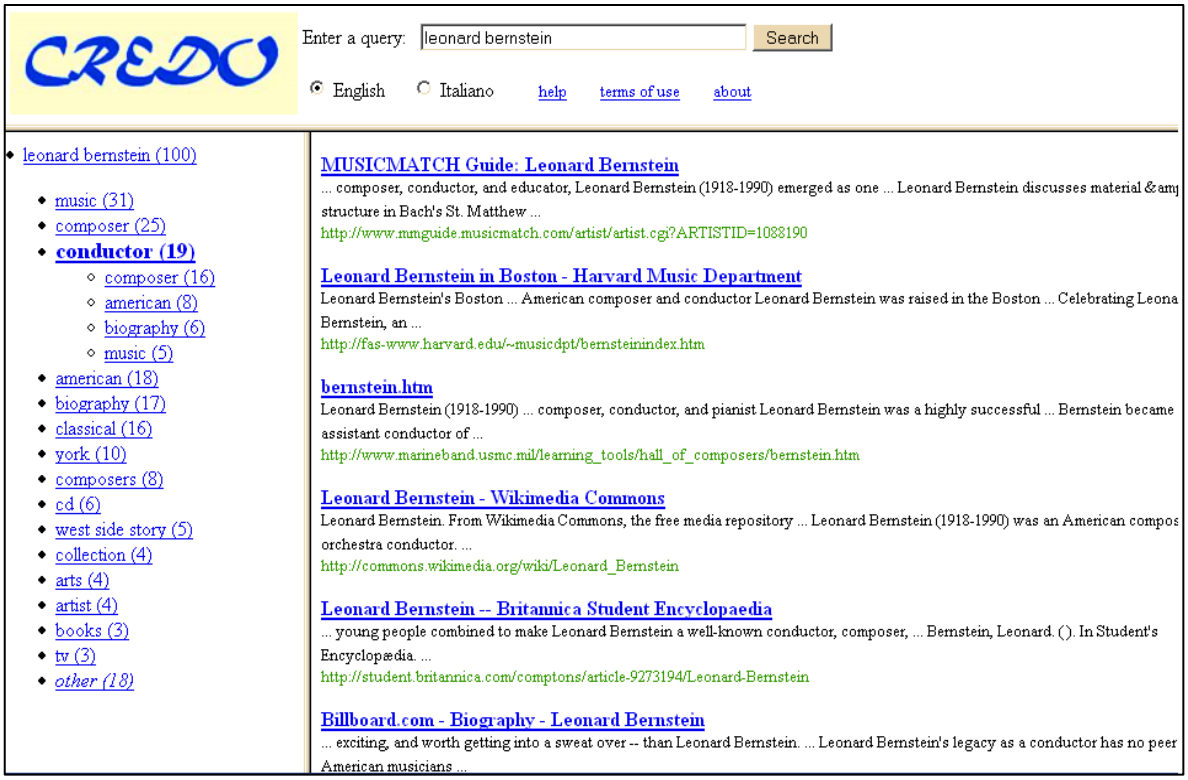


Figure 32 : *Résultats de la requête leonard + bernstein dans CREDO*

5.2 FooCA

Le principe de base de FooCA [64] est de fournir à l'utilisateur une vue d'ensemble sur les résultats retournés par un moteur de recherche. En plus, le système propose de guider et d'assister l'utilisateur lors du processus de recherche au lieu d'ignorer ses compétences humaines, telles que sa compréhension intuitive du concept formel recherché.

L'utilisateur initie la recherche en saisissant une requête et en paramétrant un certain nombre d'options. Le système évalue ces paramètres et émet la requête sans modification à un moteur de recherche (Yahoo, Google). FooCA interagit avec ce moteur via une API (*Application Programming Interface*) d'accès spécialisée mise en œuvre par ce dernier. Le moteur de recherche fournit une liste ordonnée des documents comme résultat de recherche puis FooCA, à la différence des autres systèmes, organise et visualise le résultat sous forme d'un contexte formel (voir la figure 33). A ce niveau, l'utilisateur peut encore raffiner sa recherche.

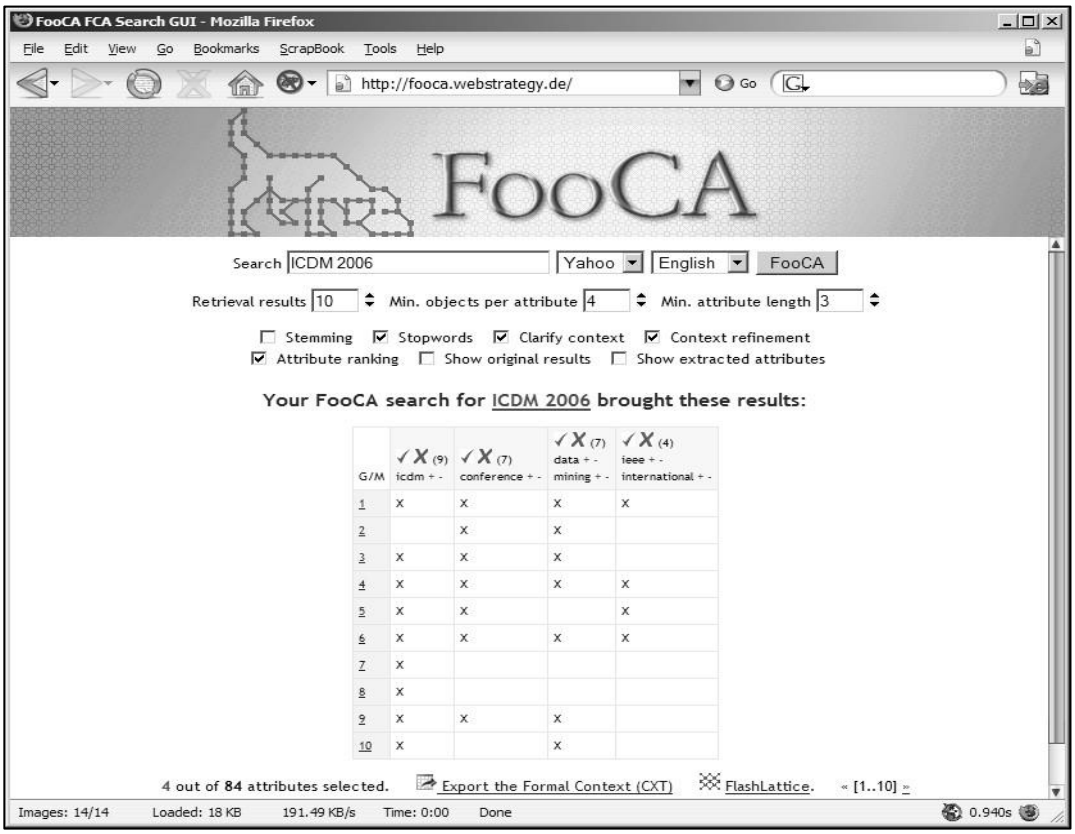


Figure 33 : *Interface du système FooCA* [64]

L'idée est d'utiliser les extraits retournés comme résultat (titres en cas d'absence d'extraits) pour extraire les descripteurs des documents sélectionnés. Après l'extraction de ces descripteurs, le système génère un contexte formel où les URLs représentent des objets et les descripteurs des propriétés. L'utilisateur peut naviguer dans le tableau représentant le contexte formel soit pour atteindre le document voulu soit pour modifier sa requête. Ainsi, grâce à un clique souris, il peut ajouter/supprimer un attribut pour mieux spécifier son besoin d'information, ou il peut lancer carrément une nouvelle requête avec de nouveaux attributs.

FooCA permet à l'utilisateur de mieux contrôler sa recherche en lui offrant la possibilité de paramétrer un certain nombre d'opérations et méthodes qui sont généralement automatiques dans la plus part des moteurs de recherche. Parmi ces fonctionnalités :

Le choix du moteur de recherche à utiliser

L'utilisateur a le choix de spécifier le moteur de recherche à utiliser ou d'utiliser plusieurs et générer le contexte formel qui correspond à chacun d'eux. Il pourra donc naviguer sur plusieurs contextes. A priori, FooCA peut intégrer n'importe quel moteur de recherche qui peut fournir une liste séquentielle des documents jugés pertinents et les extraits qui vont avec. Actuellement, les moteurs de recherche Yahoo et Google sont intégrés avec succès.

Choix de la langue de recherche

FooCA donne la possibilité de choisir entre l'anglais et l'allemand. D'autre langues peuvent être facilement intégrées, à la mesure que le moteur de recherche utilisé le permet.

Option d'indexation

Le système offre la possibilité de supprimer ou non les descripteurs correspondants aux mots vides et donne aussi une possibilité de lemmatiser. L'utilisateur peut en plus, spécifier le nombre de caractères minimum *n* des descripteurs, alors tout mot dont le nombre de caractères est inferieur à *n* va être supprimé.

Raffinement de requête par l'utilisateur

La principale différence entre le processus de raffinement de FooCA et le raffinement manuel des moteurs de recherche standards, est la liste d'attributs fournis. Typiquement, l'utilisateur n'a pas une idée claire à propos des termes qui sont nécessaires pour raffiner sa recherche. Utilisant FooCA, l'utilisateur sera en mesure d'explorer les attributs liés aux contextes des documents retournés.

Réduction du contexte

L'utilisateur peut réduire le contexte pour faciliter la visualisation de l'ensemble des résultats retournés par le moteur de recherche en augmentant le nombre d'objets pour les attributs.

Classement des attributs

Les objets sont pré-classés par le moteur de recherche. FooCA combine ce classement d'objets avec un classement d'attributs. L'objectif est de créer une zone diagonale de croix dans le contexte à partir du coin supérieur gauche au coin inférieur droit. Ainsi, faciliter la lecture du contexte. Les attributs sont classés selon le nombre d'objets auquel ils sont liés et la somme des positions de classement des objets liés.

Exportation du contexte formel

FooCA offre une interface d'exportation du contexte formel en utilisant le format Burmeister (CXT). L'utilisateur pourra donc visualiser le treillis sous son logiciel de visualisation préféré.

Visualisation de la hiérarchie de concepts formels

FooCA permet de visualiser le contexte formel sous une forme graphique. Chaque cercle représente un concept formel, et les lignes entre les concepts formels représentent leurs relations sous-concept, super-concept. Lorsque l'utilisateur clique sur un concept formel, les pages web correspondantes s'ouvrent dans de nouvelles fenêtres du navigateur (Figure 34).

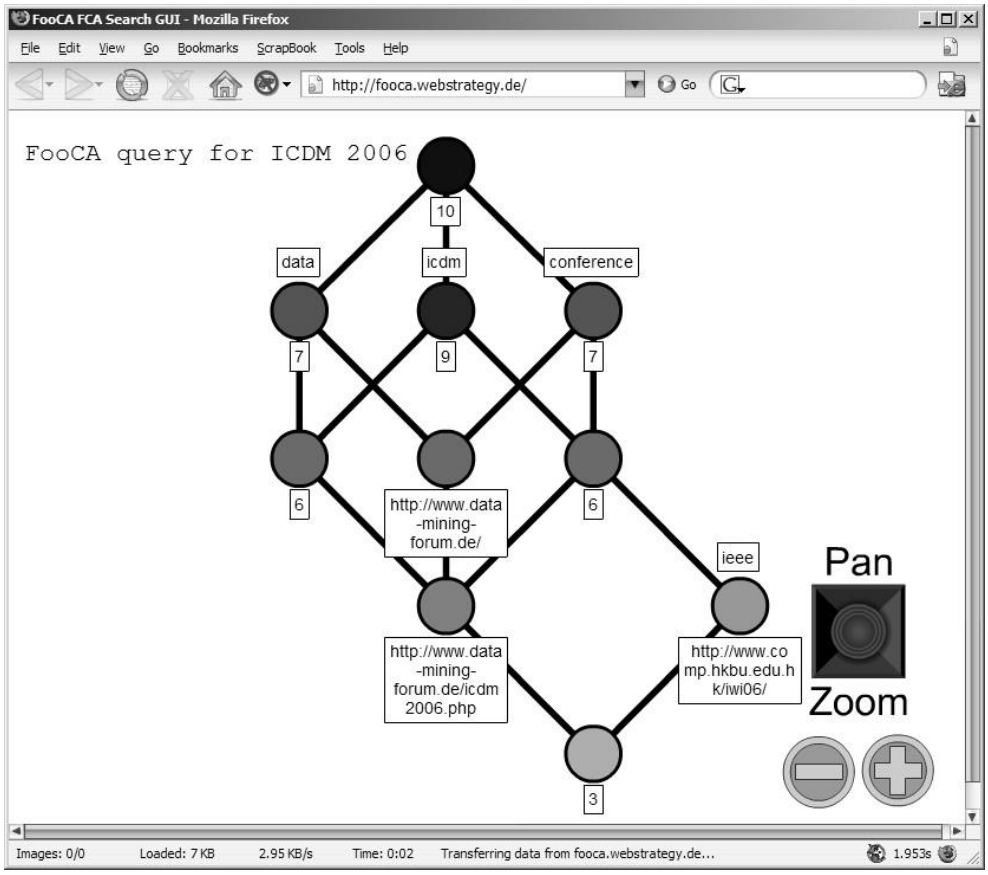


Figure 34 : *Représentation graphique de treillis dans FooCA* [26]

5.3 CreChainDo

CreChainDo [65], reprend le même principe de CREDO avec des fonctionnalités similaires à celle de FooCA. Cependant, le processus de RI proposé dans CreChainDo implémente un contrôle de pertinence explicite dans le sens où l'utilisateur peut évaluer si un concept formel du treillis est pertinent ou pas. Cela sert à modifier le contexte utilisé pour construire le treillis. Un concept formel *C* est pertinent, si l'utilisateur estime qu'une requête *Q*, formée de la conjonction de tous les mots composant l'intension de *C*, est susceptible de retourner de nouveaux documents pertinents. Un concept formel *C'*est non pertinent si l'utilisateur estime que tous les documents contenus dans son extension ne sont pas pertinents. L'architecture de CreChainDo est représentée dans la figure 35.

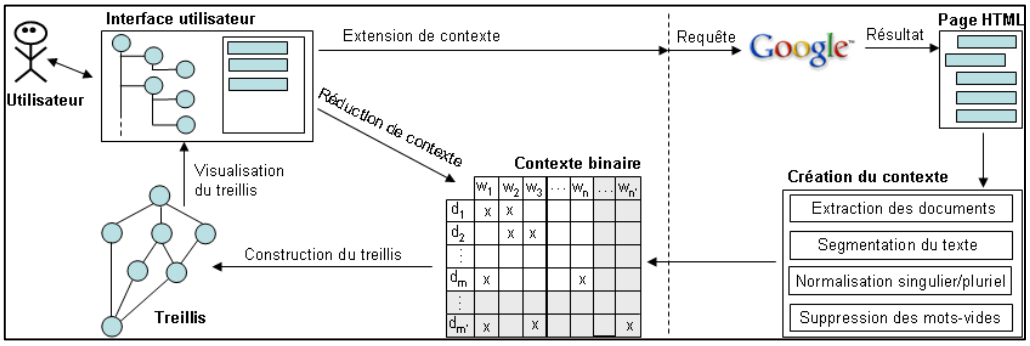


Figure 35 : *Architecture générale du système CreChainDo*

CreChainDo offre les fonctionnalités suivantes :

Rejeter un concept formel non pertinent

Pour éliminer le problème d'apparition de concepts formels non pertinents, qui résultent de la dispersion du vocabulaire des extraits des documents retournés par le moteur de recherche, CreChainDo propose de *nettoyer* la hiérarchie et éliminer le bruit en offrant la possibilité de supprimer directement les concepts formels non pertinents.

Accepter un concept formel pertinent

Dans CreChainDo, accepter un concept formel *C* permet d'étendre la sous-hiérarchie de la racine *C*. Ce qui génère des sous-concepts formels plus spécifiques que *C* par la construction d'un nouveau treillis. Ainsi, l'utilisateur peut contrôler la profondeur de la hiérarchie et le degré de spécialisation (Figure 36).

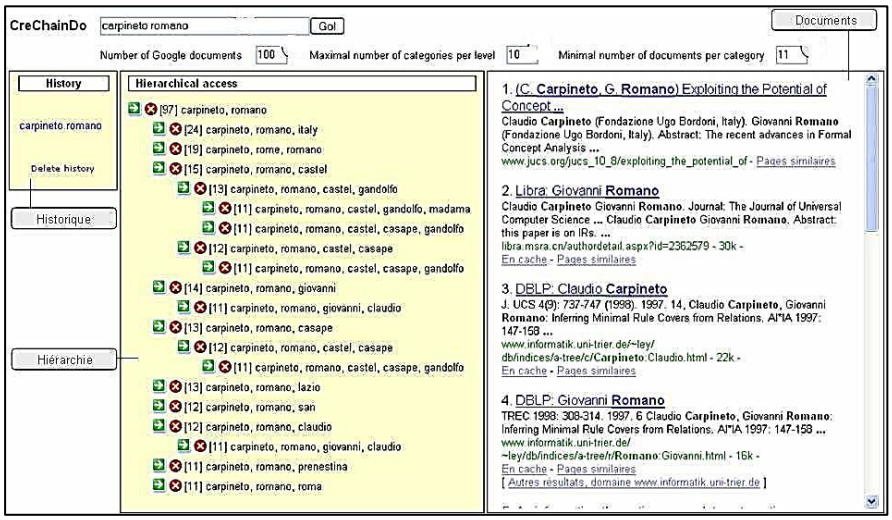


Figure 36 : *Interface de CreChainDo en réponse à une requête sur "carpineto romano"*

6 Notre système basé sur FCA pour le regroupement thématique des Résultats de recherche

La langue arabe est classée quatrième langue la plus utilisée dans le web, et le nombre de documents arabes disponibles sur le Web augmente exponentiellement. Il est donc intéressant pour les chercheurs de proposer de nouveaux systèmes de recherche d'information adaptés aux utilisateurs arabes, afin de trouver les documents pertinents. Dans cette section, nous présentons une deuxième contribution au sujet de consultation pour la langue arabe. Il s'agit d'une autre approche de regroupement thématique basée sur FCA.

6.1 Organigramme

Le système que nous proposons est décrit par un organigramme présenté dans la figure 37, et nous pouvons le résumer selon les étapes suivantes :

• Récupérer les snippets à partir de Google et de Bing.

• Prétraitement.
• Construction de Treillis.
• Trouver des clusters.
• Génération d'étiquette du cluster.

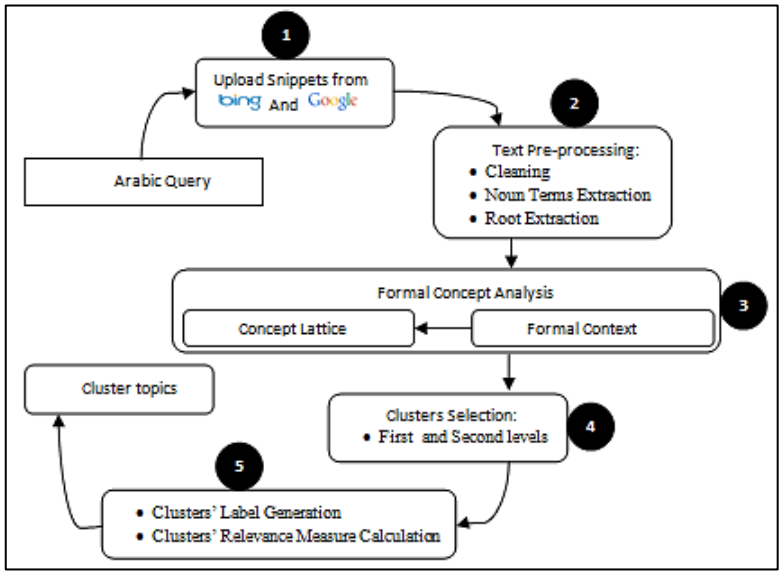


Figure 37 : *Organigramme du système proposé Basé sur FCA* [66]

6.2 Construction du Contexte Formel

L'utilisateur spécifie sa requête en langue arabe en utilisant l'interface web. La requête est envoyée aux moteurs de recherche Google et Bing à l'aide des services offerts par l'API de Google[19] et l'API de Bing[20]. La liste des résultats retournés est sous forme de snippets. Afin de rendre notre implémentation simple et facile, pour chaque snippet, on associe les quatre tags suivants (ID, URL, body et title) comme illustré sur la figure 15.

- **ID** : Identifiant du Snippet du document.
- **URL** : Lien pour accéder au contenu du document.
- **Body**: Contenu du Snippet.
- **Title**: titre de page.

Chaque snippet est nettoyé en supprimant les mots vides, les mots latins et les caractères spéciaux. La morphologie de la langue arabe nous a amené à confirmer que les termes nominatifs sont les termes les plus discriminants du contenu du document. Par conséquent, nous proposons d'ajouter un modèle linguistique pour sélectionner uniquement les termes représentant le contenu de snippet. Pour extraire les termes nominatifs, nous avons utilisé le système d'analyse morphosyntaxique arabe Al-khalil [67] mis en œuvre dans la plate-forme SAFAR[21] [130].

Ensuite, Nous cherchons pour chaque terme la racine correspondante. Finalement, les racines obtenues représentent l'ensemble des attributs, et l'ID de snippet représente l'ensemble des objets dans un contexte formel. Le tableau 22 montre un exemple de contexte formel obtenu en utilisant "الرياضة" , 'SPORT' comme requête.

[19] https://developers.google.com/custom-search/
[20] http://datamarket.azure.com/dataset/bing/searchweb
[21] http://arabic.emi.ac.ma/safar/

Dans notre cas, nous définissons les éléments du contexte formel comme suit :

• Objets : les snippets retournés par Google et Bing sans redondance, représentés par l'ID correspondant.

• Attributs : l'ensemble de racines extraites de tous les snippets.

• Relation : une relation binaire définie comme suit :

- ✓ Vraie "1" : si le mot fait partie du snippet
- ✓ Faux "0" : sinon.

6.3 Elimination des attributs redondants

L'objectif principal de cette étape est d'éliminer les informations redondantes dans le contexte formel pour produire un treillis isomorphe à l'original. À cette fin, nous proposons d'adapter la réduction des attributs redondants. Un attribut est redondant s'il a exactement les mêmes objets qu'un autre, et l'attribut de fréquence inférieure entre les deux est alors éliminé. La Figure 38 présente un exemple illustratif du processus d'élimination des attributs redondants.

Attributs / Object	a	b	c	d
1	1	0	1	1
2	1	0	0	1
3	0	1	0	0
4	1	1	0	1

Elimination des attributs redondants

Attributs / Object	a	b	c
1	1	0	1
2	1	0	0
3	0	1	0
4	1	1	0

Figure 38 : *Processus d'élimination d'informations redondantes* [66]

6.4 Construction de treillis de Gallois et génération des clusters

Le contexte formel obtenu est utilisé pour construire le concept lattice. La figure 39 montre un exemple de treillis généré en utilisant 'SPORT', 'الرياضة' comme requête d'utilisateur. Dans ce cas, nous utilisons l'API Java libre ToscanaJ, qui intègre l'algorithme de Ganter [68] pour générer l'ensemble des concepts formels et le concept lattice correspondant. Ce dernier représente un ensemble de concepts organisés en structure hiérarchique. Chaque concept regroupe un ensemble de documents (objets représentés par des ID de snippets dans des lignes de contexte formel) partageant un ensemble de termes (attributs représentés par des termes dans des colonnes de contexte formel) qui représentent l'intention. Nous notons que le concept représente un cluster lors de l'utilisation de FCA pour un processus de clustering [49].

Dans ce travail, nous proposons d'utiliser les concepts obtenus qui se produisent dans le premier et le deuxième niveau de la hiérarchie du réseau de concepts en tant que clusters obtenus. En fait, nous sélectionnons seulement le premier et le second niveau pour obtenir plus de clusters séparés avec des labels plus descriptifs. En outre, afin de faciliter la consultation pour l'utilisateur, les clusters obtenus doivent être mappés du réseau de concepts à une interface graphique et présentés en fonction de leur pertinence par rapport à la requête de l'utilisateur. À cette fin, nous avons développé une interface graphique simple dans laquelle les clusters obtenus ont été classés selon leur pertinence par rapport à la requête de l'utilisateur.

Généralement, le classement des clusters selon leurs pertinences, consiste à estimer la pertinence d'un concept qui correspond à la requête d'utilisateur. Pour surmonter ce problème, Zhang et al. [49] ont proposé une nouvelle méthode pour construire la hiérarchie des deux niveaux réduits à partir du réseau conceptuel. Cette méthode est basée sur deux mesures mathématiques [49] : la première est la mesure de l'importance du concept utilisé pour indiquer la façon dont le concept est important. Cette mesure dépend du nombre de documents dans l'extension et le nombre de concepts descendants de ce concept. La deuxième mesure est la similarité de concept, qui est basée sur le coefficient de similarité de Jaccard et sera utilisée dans le processus de fusion pour construire une hiérarchie à deux niveaux pour la consultation de l'utilisateur.

Le système proposé par Zhang et al. [49] est basé sur l'utilisation de tous les termes extraits de snippet, par conséquent, le processus de réduction est nécessaire pour réduire un certain nombre de clusters générés de façon non significative, en filtrant ou en regroupant des concepts similaires. Dans notre système, on utilise les racines des termes nominatifs au lieu de tous les termes, ce qui rend la réduction une étape inutile. En fait, le nombre des termes nominatifs dans chaque snippet est réduit, et ces termes sont liés au sujet du document correspondant. Cependant, l'estimation de la pertinence d'un concept correspondant à la requête d'un utilisateur est nécessaire pour faciliter le processus de consultation pour ce dernier. Comme nous l'avons mentionné précédemment, un concept se caractérise par deux composantes : l'extension et l'intention.

Pour améliorer la performance de notre système, nous proposons une nouvelle mesure de pertinence de concept qui prend en considération les deux composantes suivantes :

- Le nombre de documents dans l'extension.
- Le poids de chaque mot dans l'intention.

Nous définissons notre pertinence proposée S(Ci) comme une mesure du concept Ci comme suit :

$$\text{Extent\_Weight} = (|\text{Extent}(Ci)| / \text{Nbr\_Total\_Docs}) \quad \textbf{(4.3)}$$

$$\text{Intent\_Weight} = \sum (\text{TF.IDF}(\text{Intent}(Ci)) / |\text{Intent}(Ci)| \quad \textbf{(4.4)}$$

$$S(Ci) = \text{Extent\_Weight} * \text{Intent\_Weight} \quad \textbf{(4.5)}$$

où :

- |Extent(ci)|: Le nombre de documents dans l'extension.
- Nbr_Total_Docs: le nombre total de snippets dans le corpus.
- $\sum(TF.IDF(Intent(Ci))/|Intent(Ci)|$: La moyenne de TF.IDF «*Term Frequency–Inverse Document Frequency* **»** de tous les mots de l'intention dans le concept correspondant.

6.5 Génération d'étiquette du cluster

La génération de label de clusters est une étape cruciale, car les labels sans signification ou trompeuses peuvent amener les utilisateurs à vérifier des clusters erronés. En outre, les labels doivent être plus compréhensibles pour l'utilisateur et décrivent avec précision le contenu des documents. À cette fin, nous proposons de trouver le terme original de chaque élément dans l'intention du concept correspondant, et l'étiquette du cluster constituée des termes originaux séparés par des virgules. Ensuite, l'utilisateur peut simplement cliquer sur l'étiquette du cluster qui décrit mieux son besoin.

6.6 Exemple illustratif

Dans cette section, vu le grand nombre de pages retournées par les moteurs de recherche et par souci de simplicité, nous présentons un exemple illustratif en utilisant sept snippets de la première page renvoyée par Google et Bing en utilisant (الرياضة) comme requête (Tableau 22) pour expliquer comment notre système fonctionne et facilite le processus de consultation.

ID	Snippets
1	Yahoo! حصاد 2012 الرياضي - مكتوب الرياضي -
2	اخبار الرياضة ومباريات اليوم من يوروسبورت عربية
3	اخبار الرياضة وكرة القدم - أخبار سكاي نيوز عربية
4	أخبار الرياضة _ رياضة دوت كوم
5	مكتوب Yahoo! أخبار الرياضة | مكتوب الرياضي | آخر الأحداث الرياضية –
6	الجزيرة الرياضية: الأخبار
7	الرياضة - روسيا اليوم

Tableau 22 : *Exemple de sept snippets de la requête (الرياضة)*

Chaque snippet est nettoyé en supprimant les mots vides. Ensuite, nous allons chercher la racine correspondante pour chaque terme dans le snippet. Les racines obtenues représentent l'ensemble des attributs dans le contexte formel et les identificateurs de snippets représentent l'ensemble des objets. Le tableau 23 montre le contexte formel obtenu de ces sept snippets correspondant à la requête (الرياضة). Notons que dans l'exemple du tableau 23, les termes sont représentés dans la forme initiale sans la phase de racinisation.

	حصاد	الرياضة	مكتوب	الخبر	عربية	بروروسبورت	كرة	القدم	سكاي	دوت	اخر	الاحداث	الجزيرة	روسيا
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	1	1	1	0	0	0	0	0	0	0	0
3	0	1	0	1	1	0	1	1	1	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0	1	0	0	0	0
5	0	1	1	1	0	0	0	0	0	0	1	1	0	0
6	0	1	0	1	0	0	0	0	0	0	0	0	1	0
7	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Tableau 23 : *Contexte formel de la requête (الرياضة)*

Dans cette étape, nous éliminons les attributs redondants dans le tableau (23) de contexte formel. Le tableau (24) présente le contexte formel après le processus d'élimination des attributs redondants.

	حصاد	الرياضة	مكتوب	الخبر	عربية	بروروسبورت	كرة	دوت	اخر	الجزيرة	روسيا
1	1	1	1	0	0	0	0	0	0	0	0
2	0	1	0	1	1	1	0	0	0	0	0
3	0	1	0	1	1	0	1	0	0	0	0
4	0	1	0	1	0	0	0	1	0	0	0
5	0	1	1	1	0	0	0	0	1	0	0
6	0	1	0	1	0	0	0	0	0	1	0
7	0	1	0	0	0	0	0	0	0	0	1

Tableau 24 : *Contexte Formel après le processus d'élimination des Attributs redondants*

Le contexte formel présenté dans le tableau 24 est utilisé d'une part pour construire le Treillis présenté dans la figure 39, et d'autre part pour obtenir les clusters, qui sont présentés dans une structure hiérarchique à différents niveaux. Comme nous remarquons sur la figure 39, il y a trois concepts dans le premier niveau qui représente les concepts les plus généraux :

- **(2, 3, 4, 5, 6; الرياضة, الخبر)**
- **(1, 5; الرياضة, مكتوب)**
- **(7, الرياضة, روسيا)**

L'utilisateur choisit de parcourir l'un des clusters du premier niveau en cliquant sur leurs labels, ensuite, l'utilisateur peut accéder à plus de thèmes au deuxième niveau.

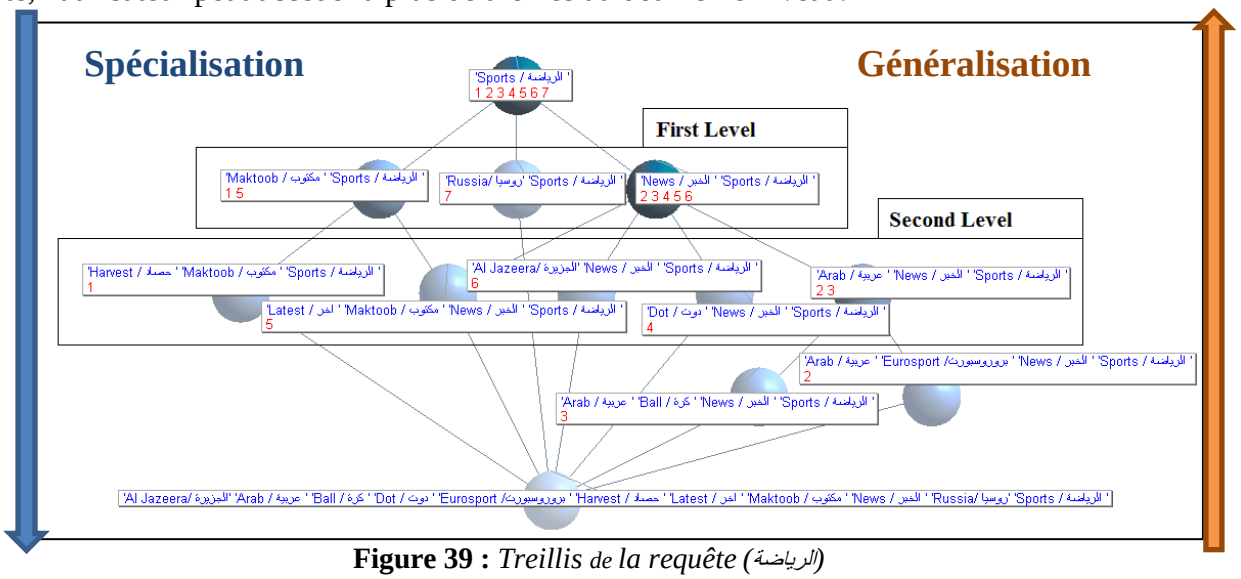


Figure 39 : *Treillis de la requête (الرياضة)*

Afin d'améliorer le processus de consultation dans le cadre de ce travail, nous proposons de classer et présenter les clusters en fonction de leurs pertinences. L'exemple de la figure 40 présente trois clusters dans le premier niveau d'hiérarchie obtenue en utilisant la requête « ***الرياضة***». De plus, tous les labels de clusters obtenus sont compréhensibles. Nous notons également que le premier cluster, qui est marqué par (الرياضة, اخبار) est le plus pertinent à la requête (الرياضة). Le nombre affiché à côté du label correspond au nombre de documents similaire qui forment ce cluster.

Figure 40 : *Interface web pour visualiser les clusters obtenus* [66]

7 Résultat d'expérimentation et discussion

Dans cette section, pour mettre en évidence l'intérêt de notre contribution, nous proposons une étude comparative de notre système avec deux autres. Il s'agit du système basé sur STC et celui nommé Lingo. Les deux systèmes sont intégrés dans la plate-forme carrot^2, qui est un moteur de clustering de résultats de recherche open source. Nous notons que la version 3.2.0 de carrot^2 introduit un support expérimental pour le clustering du contenu en arabe. Cette étude comparative s'intéresse à la qualité des résultats du regroupement thématique et les labels produits par différents systèmes.

Dans cette étude comparative, nous avons utilisé Open Directory Project (ODP) qui est un annuaire multi-langue, composé de quelques millions de pages Web pré-classées et organisées en arborescence. Pour la langue arabe, l'ODP inclut 4781 snippets pré-classés en 459 catégories par un groupe d'experts. Par conséquent, l'ODP représente un bon corpus de test pour notre étude comparative.

7.1 La qualité de clustering

Généralement, la qualité des résultats de regroupement thématique de n'importe quel système de regroupement thématique peut être mesurée par le degré auquel ce système est capable de regrouper un ensemble de snippets pré-classés dans les mêmes catégories sans connaître l'attribution de catégories d'origines.

La qualité des résultats du regroupement peut être mesurée à l'aide de deux mesures : « ***Normalized Mutual Information*** » (NMI) and « ***Normalized Complementary Entropy*** » (NCE)

[69]. Ces paramètres sont employés par Geraci et al. [70] pour mesurer la qualité et la performance des différents algorithmes de regroupement thématique.

Pour un ensemble donné S de N snippets pré-classés sous C = {c1, c2, ... cn} de catégories et un ensemble C '= {c'1, c'2, ... c'm} comme résultats du clustering, le NMI et le NCE sont définis comme suit [70]:

$$NMI(C,C') = \frac{2}{\log|C||C'|}\sum_{c\in C}\sum_{c\prime\in C\prime} P(c,c')\log\frac{P(c,c')}{P(c)P(c')} \quad \textbf{(4.4)}$$

où : $P(c) = \frac{|c|}{N}, P(c') = \frac{|c\prime|}{N}, P(c,c') = \frac{|c\cap c\prime|}{N}$

$$NCE(C,C') = \sum_{i=1}^{m}\frac{|c_i{}'|}{N'}NCE(C,c_i{}') \quad \textbf{(4.5)}$$

où : $NCE(C,c_i') = 1 - \frac{2}{\log|C|}\sum_{j=1}^{n} -\frac{P(c_j,c_i')}{P(c_j)}\log\frac{P(c_j,c_i')}{P(c_j)}$ et $N' = \sum_{i=1}^{m}|c_i{}'|$

NMI est conçu pour évaluer le non-chevauchement de clustering, donc les valeurs les plus élevés de NMI indiquent une meilleure qualité de clustering.

NCE varie dans l'intervalle [0, 1] a été conçu pour évaluer le chevauchement des clusters similaires. Une plus grande valeur de NCE signifie une meilleure qualité de clustering. Zhang et al. [49] montrent que ces métriques souffrent des problèmes suivants :

• Pour un nombre des clusters d'origine K dans le corpus de test, plus le nombre des clusters générés K^* augment, plus les valeurs obtenues de NMI et NCE augmentent.

• Pour un nombre des clusters générés K, plus le nombre des clusters générés K^* des clusters d'origine augment, plus que la valeur de NMI obtenu augmente.

• Lors de la comparaison des résultats obtenus par deux algorithmes de regroupement thématique différents avec NCE et NMI, la performance peut être inversée si nous changerons différentes clustres d'origine.

Pour surmonter les problèmes cités ci-dessus des deux métriques, nous proposons deux métriques améliorées : A-NMI @ K et A-NEC @ K, où A indiqué la moyenne et K indique le nombre de clusters utilisés dans l'expérience. Dans cette étude comparative, nous prenons différentes valeurs de K comme 10, 20 séparément.

La figure 41 présente les valeurs A-NMI @ 10, A-NMI @ 20 et A-NMI @ ALL pour mesurer la qualité de non chevauchants des clusters des trois systèmes : notre système, STC et Lingo. Il est clair que notre système produit des résultats meilleurs que les deux autres et apporte une amélioration deux fois meilleure comparé aux deux autres. La figure 42 présente les valeurs A-NCE @ 10, A-NCE @ 20 et A-NCE @ ALL pour mesurer la qualité de chevauchement de clusters, et elle montre que notre système présente des résultats meilleurs que les deux autres.

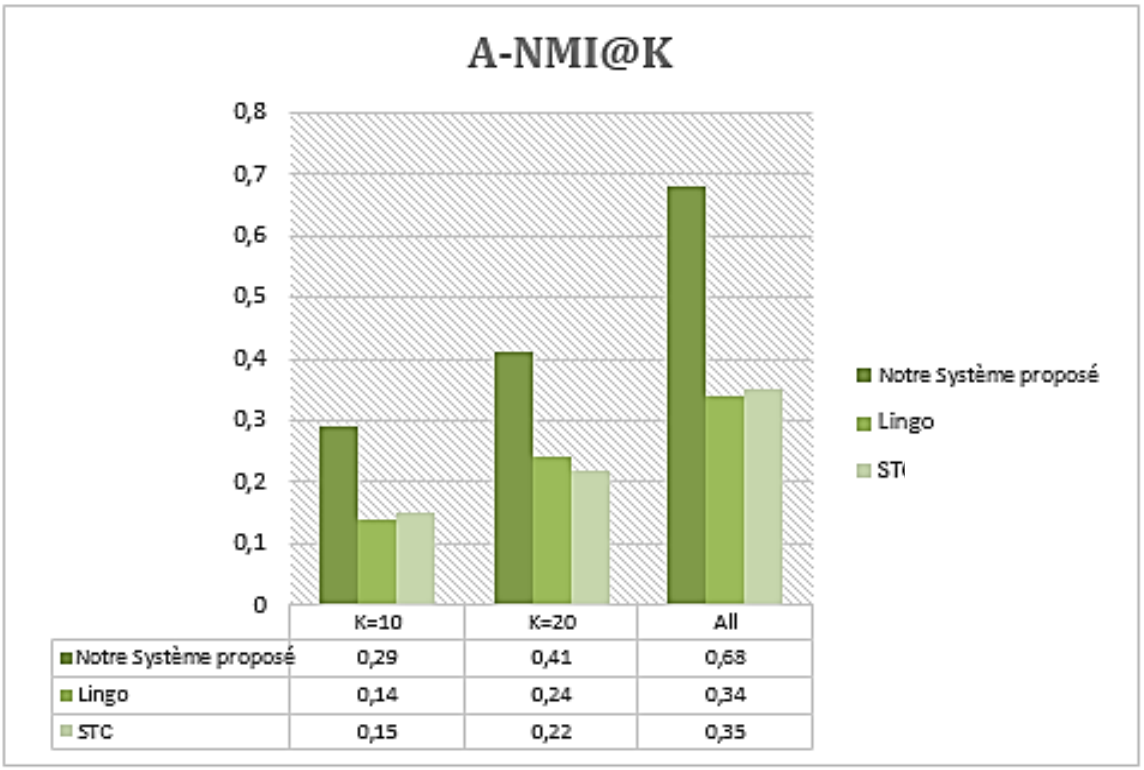


Figure 41 : *Les valeurs A-NMI@K*

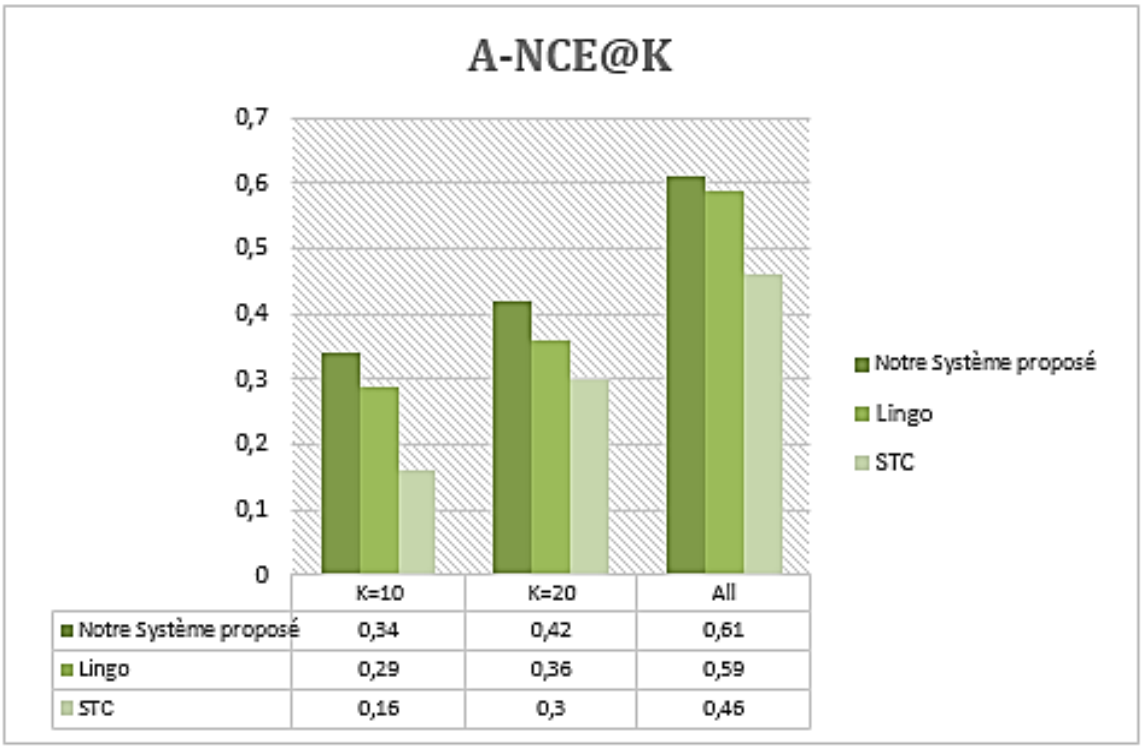


Figure 42 : *Les valeurs A-NCE@K*

7.2 Qualité du label de cluster

Le label d'un cluster doit bien décrire le contenu du cluster. Un mauvais label qui ne reflète pas le contenu du cluster va empêcher l'internaute de le consulter même si il contient des documents pertinents pour lui. Le but principal est de comparer la qualité du label d'un cluster pour les trois

systèmes. Dans cette section, pour mettre en évidence l'intérêt de notre proposition au niveau de la génération du label de cluster, nous proposons de mener une étude d'évaluation et de comparaison de notre système avec les deux autres systèmes les plus connus (STC, Lingo). Généralement, l'évaluation par des experts n'est pas toujours possible, en raison du manque de ressources humaines. En outre, les experts ne peuvent pas évaluer plusieurs milliers de requêtes utilisateurs sur système en ligne de regroupement thématique des résultats de recherche web. Par conséquent, nous présentons ici deux exemples requête utilisateur pour avoir une idée de la qualité de label de cluster pour chaque système. La première est (تجارة) et la seconde est (التعليم).

Notre Système proposé	Lingo	STC

Figure 43 : *Labels générés par les 3 systèmes en utilisant la requête (تجارة)*

Notre système proposé	Lingo	STC

Figure 44 : *Labels générés par les 3 systèmes en utilisant la requête (التعليم)*

La figure 43 et la figure 44 affichent les labels des 10 clusters produits par notre système, Lingo et STC pour les deux requêtes. Selon des utilisateurs, les étiquettes produites par les trois systèmes sont lisibles et informatives. Cependant, chaque cluster généré par Lingo ou STC contient peu de documents, le nombre indiqué par le nombre suivi de chaque label, ce qui signifie que de nombreux documents ne sont pas regroupés dans les 10 clusters du système Lingo et STC. D'autre part, les labels produits par notre système ne sont pas basés sur des phrases, mais sur des mots-clés. Chaque label est composé d'un ou plusieurs mots clés. Notre système permet de découvrir la relation entre les termes dans l'ensemble des résultats. La figure 43 montre aussi qu'il y a un problème dans notre système lié à l'étape de prétraitement. Il y a les deux labels de classe شركة et شركات qui ont la même racine et qui normalement devraient être fusionnées. En général, les résultats présentés dans les deux requêtes ont prouvé l'efficacité de notre système par rapport aux deux autres.

8 Conclusion

Dans ce chapitre, nous avons proposé une deuxième contribution apportée au sujet de la consultation. En effet, nous avons utilisé l'Analyse Formels de Concepts dans un nouveau système de regroupement thématique de résultats de recherche web pour la langue arabe. Le système proposé regroupe automatiquement les résultats de recherche retournés en clusters d'une part de haute qualité avec une structure hiérarchique, et d'autre part fournit des labels descriptifs des clusters. Une série d'expériences a été menée : des évaluations subjectives et objectives ont été présentées en utilisant les snippets retournés par les moteurs de recherche Google et Bing. Les résultats obtenus ont été très encourageants et illustrent la performance de notre système proposé basé sur FCA par rapport à deux autres systèmes.

Conclusion de la partie 2

La consultation des résultats de recherche est l'un des principaux problèmes auxquels font face les moteurs de recherche Web traditionnels (Google, Yahoo et Bing) pour l'anglais, et les autres langues en général et pour l'arabe en particulier. L'organisation des résultats de recherche en arabe sous forme de clusters facilite aux utilisateurs arabes l'accès rapide à ces résultats.

Dans cette partie, nous avons présenté deux contributions pour remédier au problème de consultation. La première contribution est présentée dans le chapitre 3, où nous avons proposé d'intégrer l'algorithme STC dans notre système de regroupement de résultats de recherche. Dans le chapitre 4, nous avons présenté la deuxième contribution apportée au sujet de consultation. En effet, nous avons utilisé l'analyse de concepts formels dans un nouveau système de regroupement thématique de résultats de recherche web pour la langue arabe. Les résultats obtenus illustrent la performance de notre système proposé basé sur FCA par rapport aux autres systèmes.

Partie 3 : Problème d'indexation

- **Chapitre 5 : « KpST » et « Improved KpST » systèmes pour les extractions des phrases-clés**
- **Chapitre 6 : Nouvelle Méthode d'indexation basée sur les phrases-clés**

Introduction de la partie 3

Le processus d'indexation se considère comme une étape hors-ligne primordiale dans n'importe quel système de recherche d'information, et qui peut influencer d'une façon positive ou négative la performance de résultat d'un tel système SRI.

Dans un environnement de la recherche d'information, l'index sert comme une indication de pertinence du document. Comme les phrases-clés reflètent les principaux thèmes d'un document, ils peuvent être utilisés pour regrouper des documents en mesurant le chevauchement entre les phrases-clés qui leur sont assignées. Les phrases-clés peuvent également être utilisées de manière proactive dans l'indexation pour la recherche d'information.

Dans cette partie, nous présentons dans le chapitre 5 les différentes méthodes que nous recensons pour effectuer la tâche d'extraction des phrases-clés à partir des documents en langue arabe et nous détaillons notre nouvelle approche non supervisée pour l'extraction des phrases-clés à partir des documents textuels arabes. Cette dernière est basée sur la structure de données d'arbre de suffixes. Nous présentons ensuite dans le chapitre 6 le processus d'indexation. Nous commençons par citer l'ensemble des problèmes liés à l'utilisation d'indexation basée sur les mots-clés simples sur les SRI en particulier pour la langue arabe, ainsi que les solutions présentées dans la littérature pour surmonter ces problèmes. Par la suite, nous présentons notre SRI dédié à la langue arabe basé sur le concept des phrases-clés.

CHAPITRE 5. « KpST » et « Improved KpST » les systèmes pour l'extraction des phrases-clés

1 Introduction

Les phrases-clés sont définies comme des phrases qui captent les principaux thèmes abordés dans un document. Généralement dans le domaine de la recherche d'information, les phrases-clés peuvent être trouvées dans la plupart des bibliothèques numériques et des systèmes de Recherche d'Information [71]. La plupart des informations sur le Web sont sous une forme textuelle, et elles se développent avec un rythme rapide. Pour un grand volume de documents, il sera inefficace pour des experts d'extraire manuellement les phrases-clés. Par conséquent, la nécessité de l'automatisation des processus pour extraire des phrases-clés à partir de documents textes. L'extraction des phrases-clés est une tâche difficile dans le traitement du langage naturel [72].

Les phrases-clés peuvent être utilisées pour diverses applications. *Turney* [71] énumère plus d'une douzaine d'applications qui utilisent l'extraction des phrases pertinentes. Par exemple, en fournissant des mini-résumés des documents volumineux, soulignant des phrases dans le texte, la compression de texte, le raffinement des requêtes, le clustering et la classification des documents. Les algorithmes d'extraction des phrases pertinentes se répartissent en deux catégories : l'extraction des phrases pertinentes des documents textes, qui est souvent posée comme une tâche d'apprentissage supervisée et l'extraction des phrases pertinentes d'un ensemble de documents, ce qui est une tâche d'apprentissage non supervisée qui tente à découvrir les sujets plutôt que d'apprendre à partir des exemples.

Dans un environnement de la recherche d'information, l'index sert comme une indication de pertinence du document pour les utilisateurs. Comme les phrases-clés reflètent les principaux thèmes d'un document, ils peuvent être utilisés pour regrouper des documents en mesurant le chevauchement entre les phrases-clés qui leur sont assignées. Les phrases-clés peuvent également être utilisées de manière proactive dans la Recherche d'Information, pour l'indexation. En dépit de ces avantages connus des phrases pertinentes, seule une minorité de documents ont des phrases pertinentes qui leur sont assignées. C'est parce que les auteurs fournissent des phrases seulement quand ils sont chargés de le faire [73]. L'assignation manuelle des phrases est coûteuse et prend du temps. C'est ce besoin intense qui a motivé la recherche à trouver des approches automatisées pour l'extraction des phrases pertinentes.

De l'analyse de phrases pertinentes construites par l'être humain, *Z. Liu et al.* [74] , nous avons conclu que les bonnes phrases d'un document doivent satisfaire les propriétés suivantes :

- Compréhensibles : Les phrases sont compréhensibles pour les gens. Ceci indique que les phrases extraites devraient être grammaticalement correctes. Par exemple, *"machine learning"* est une phrase grammaticalement correcte, mais pas *"machine learned"*.

- Pertinentes : Les phrases sont sémantiquement pertinentes avec le thème du document. Par exemple, pour un document sur *"machine learning"*, nous voulons que les phrases extraites correspondent toutes au thème de *"machine learning"*.
- Une bonne couverture : Les phrases doivent bien couvrir l'ensemble du document. Supposons que nous ayons un document décrivant *«Beijing»* à partir de divers aspects de la *"localisation"*, *"l'atmosphère"* et *"la culture"*, les phrases extraites devraient couvrir ces trois aspects.

De nombreux efforts remarquables ont été proposés et mis en œuvre pour extraire automatiquement des phrases pertinentes de documents en anglais et en d'autres langues [75]. En revanche, peu d'efforts sont réalisés pour les documents écrits en langue arabe. Bien que certains chercheurs aient appliqué leur système d'extraction des phrases pertinentes de documents en arabe, l'efficacité prouvée des phrases extraites n'était pas satisfaisante.

2 L'extraction des phrases-clés pour les documents textes arabes

Cette section donne un état d'art des approches existantes pour l'extraction des phrases- clés pour les documents de texte Arabe. Généralement, les algorithmes d'extraction de phrases clés proposés peuvent être classés dans deux catégories : Approches supervisées et approches non supervisées.

2.1 Approches supervisées

El-shishtawy et Al-sammak [75] présentent une technique d'apprentissage supervisée pour extraire les phrases pertinentes des documents textuels arabes. L'extracteur est fourni avec des connaissances linguistiques pour améliorer son efficacité au lieu de compter uniquement sur les informations statistiques telles que la fréquence des termes et les distances. Lors de l'analyse, un corpus Arabe annoté est utilisé pour extraire les caractéristiques lexicales requises des mots du document. La connaissance comprend également des règles syntaxiques fondées sur une partie de balises vocales et des séquences de mots autorisés à extraire les phrases candidates. Dans le travail de T.A.El-shishtawy et al. [75], les stems des mots arabes sont utilisés à la place des racines pour représenter les termes candidats. Le stem garde la plupart des inflexions trouvées dans le mot arabe.

Duwairi *et al.* [76] présentent le système d'extraction de phrases pertinentes *«Shihab»* à partir de documents arabes. *Shihab* considère l'extraction de phrases pertinentes comme un problème de classement. La liste des phrases pertinentes est générée par le clustering des phrases d'un document. Les phrases sont construites à partir de mots qui apparaissent dans le document. Ces phrases se composent de 1, 2 ou 3 mots. L'idée est de grouper les phrases qui sont similaires dans le même cluster. La similarité entre les phrases est déterminée en calculant la valeur de Dice de leurs contextes correspondants. Un contexte de phrase est la partie dans laquelle cette phrase apparaît. Le Clustering

agglomératif hiérarchique est utilisé dans la phase de clustering. Une fois que les clusters sont prêts, chaque cluster nomme une phrase à l'ensemble des phrases candidates. Cette phrase est appelée représentant du cluster et est déterminée en fonction d'un ensemble d'heuristiques.

Nous signalons que les méthodes citées précédemment pour l'extraction des phrases pertinentes sont des méthodes basées sur l'approche supervisée qui nécessitent une étape d'apprentissage, ce qui affecte la qualité des résultats dans le cas de l'utilisation d'un corpus d'apprentissage insuffisant.

2.2 Approches non supervisées

Le système KP-Miner proposé par El-Beltagy et al. [77] peut être considéré comme l'un des systèmes les plus connus pour l'extraction des phrases clés en particulier pour la langue arabe. Le système KP-Miner peut se résumer en trois étapes logiques :

La sélection des candidats des phrases-clés

Chaque séquence de mots sera une ***phrase-clé*** candidate si elle vérifie les conditions suivantes :

- La séquence de mots n'est pas séparée par des signes de ponctuation ou des mots vides.
- La séquence de mots doit apparaître au moins n fois dans le document, où n est le facteur de fréquence le moins vu afin d'être considéré comme une ***phrase-clé*** candidate. Ce facteur sera décrémenté si le document est court. Généralement, il est fixé à trois pour l'anglais et deux pour l'arabe.
- La troisième condition est liée à la position où la ***phrase-clé*** apparaît pour la première fois dans le document d'entrée. Par l'observation, ainsi que par l'expérimentation, il a été constaté que, une phrase qui apparaît pour la première fois dans les documents longs après une position donnée est très rarement une phrase-clé, elle est donc filtrée et ignorée.

Calcul du poids de chaque candidat

En regardant presque tous les documents, on peut observer que l'apparition de phrases est beaucoup moins fréquente que l'apparition des termes simples dans le même document. On peut donc conclure que l'une des raisons pour lesquelles TF-IDF effectue une mauvaise tâche lorsqu'elle est appliquée à la tâche d'extraction des phrases-clés, est le fait qu'elle ne prenne pas en compte ce fait qui aboutit à un biais vers des mots simples car ils se produisent en grand nombre. Par conséquent, un facteur de renforcement est nécessaire pour les termes composés afin d'équilibrer ce biais vers des termes uniques. Chaque candidat w_{ij} est scoré comme suit :

$$w_{ij} = tf_{ij} * idf * B_i * P_f \quad \textbf{(5.1)}$$

- ✓ w_{ij} est le score du terme t_j dans le document d_i
- ✓ tf_{ij} est la fréquence du terme t_j dans le document d_i
- ✓ idf est log 2N/n où N est le nombre de documents dans la collection et n est le nombre de documents où le terme t_j se produit au moins une fois.

- ✓ Bi est le facteur de relance associé au document d_i,
- ✓ P_f est le terme facteur associé à la position.

Raffinement

L'utilisateur spécifie un nombre n de phrases-clés qui veut comme résultat sur une liste triée. La valeur par défaut de n est 5.

Dans le cadre de cette thèse, pour traiter le problème d'indexation des documents de textes arabes, nous proposons une nouvelle approche non supervisée basée sur l'utilisation des arbres de suffixes. Cette approche sera présentée et détaillée dans la section suivante.

3 KpST : Nouvelle méthode proposée pour l'extraction des phrases-clés

Dans cette section, une nouvelle méthode non supervisée d'extraction des phrases pertinentes basée sur la construction de l'arbre des suffixes généralisé pour les documents en arabe est présenté [78]. Le système que nous proposons est décrit par l'organigramme suivant (Figure 45) :

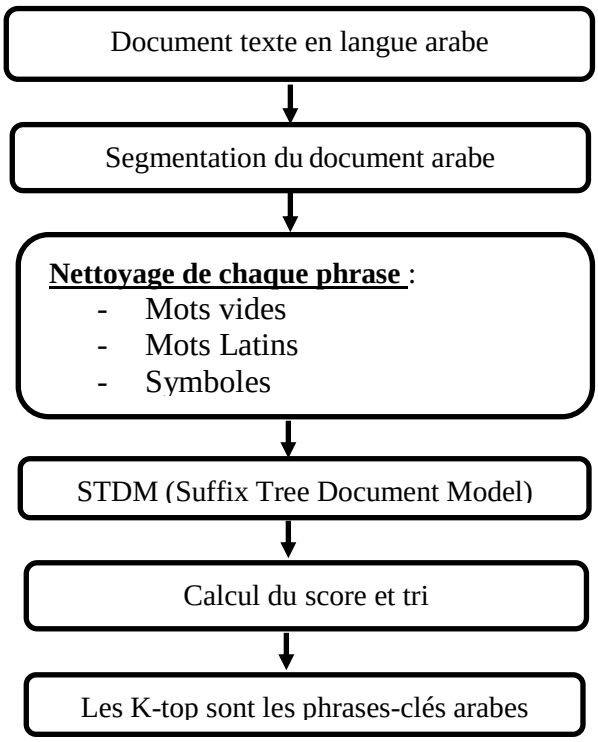


Figure 45 : *Description de notre nouvelle approche KpST*

3.1 «Nettoyage» du document

Dans cette étape, nous supprimons les mots vides de chaque document comme (فانه, وان, والذي, وهذا, فكان, ستكون ...). En outre, dans notre cas, pour faire face notamment aux documents textes arabes, nous proposons également dans cette étape de supprimer les mots latins et les caractères spéciaux.

3.2 Modèle de l'Arbre des Suffixes d'un Document

Comme nous l'avons mentionné précédemment, le *Modèle de l'Arbre des Suffixes d'un Document (Suffix Tree Document Model (STDM))* considère un document $d = w_1, w_2, \ldots w_m$ comme une chaîne constituée de mots w_i *(tel que i = 1 , 2, ... , m).*

Un arbre des suffixes du document *d* est un arbre compact contenant tous les suffixes du document *d*. Cet arbre est présenté par un ensemble de nœuds, des feuilles et des étiquettes. L'étiquette d'un nœud de l'arbre est définie comme étant la concaténation, dans l'ordre, des sous-chaînes d'étiquetage des bords du chemin de la racine à ce nœud. Chaque nœud doit avoir un score, et le nœud peut être classé en fonction de son score.

La pertinence dépend de :

- La longueur de l'étiquette du nœud.
- Le nombre d'occurrences du mot dans le document (*Term Frequency*).

Chaque nœud de l'arbre des suffixes est scoré comme suit :

$$S(B) = |B| \times F(|P|) \times \sum_i^n TF.IDF(w_i) \quad \textbf{(5.2)}$$

$$F(|P|) = \begin{cases} |P|, \text{ if } 3 \geq |P| \geq 1 \\ 0, \text{ otherwise} \end{cases} \quad \textbf{(5.3)}$$

où

- $|B|$ est le nombre de répétitions de la phrase P dans le document présenté par le nœud *B*,
- $|P|$ est le nombre de mots qui composent la phrase P
- w_i représente les mots dans la phrase P
- $TFIDF(w_i)$ représente la fréquence du terme et la fréquence inverse du document pour chaque mot w_i dans la phrase P.

4 Amélioration de KpST « Improved KpST »

L'utilisation du système KpST [78] pour extraire des phrases clés sur des documents textes Arabes prouvent la puissance de ce système [79], [80]. Nous notons que ce système sera intégré dans d'autres applications de fouille de texte, de sorte que sa durée d'exécution a un impact très important sur leurs performances. Cependant, la morphologie de la langue arabe nous a amenée à confirmer que les phrases clés sont sous la forme des phrases nominales en arabe. En outre, comme nous le savons, la plupart des systèmes de phrases clés souffrent du problème de sous-phrases-clés. Généralement, chaque phrase clé générée peut contenir un certain nombre sous phrases clés qui lui font partie. Par conséquent, l'objectif est de garder seulement les phrases clés les plus pertinentes. Nous proposons d'adapter une nouvelle architecture pour notre système KpST, en ajoutant deux nouvelles couches. La Figure 46 décrit notre système Improved KpST [81].

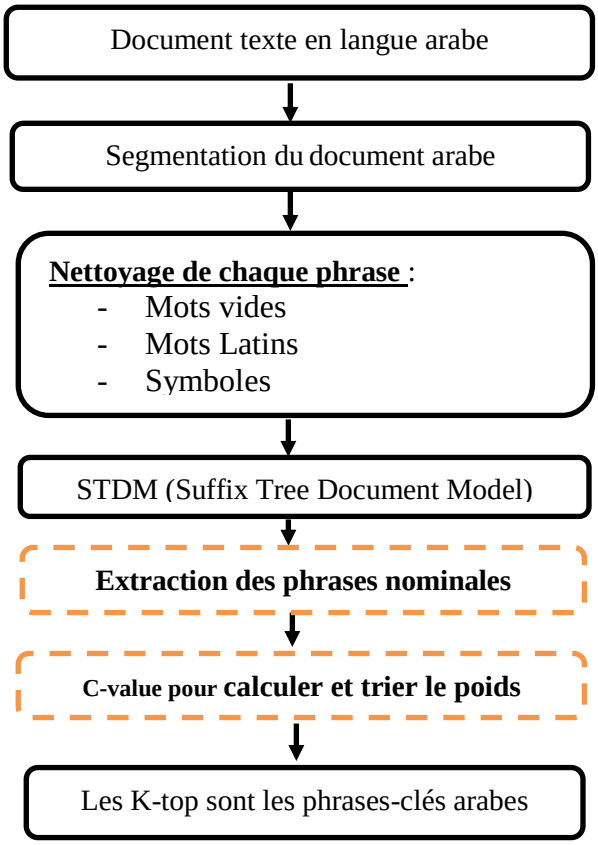


Figure 46 : *Description de « Improved KpST »* [81]

4.1 Extraction des phrases nominales

La phrase se compose d'unités de mots qui comprennent son sens. En arabe il y'a deux sortes de phrases : la phrase verbale, qui commence par un verbe, par exemple (يأكل القط الفأر / Le chat mange la souris), et la phrase nominale, qui commence par un nom, par exemple (الفأر يأكل الجبن / La souris mange le fromage), cette dernière se compose de deux parties :

- ✓ Le sujet (المبتدأ), Il peut s'agir d'un nom oud'un pronom.
- ✓ Le prédicat (الخبر), nous dit quelque chose sur le sujet. Il peut s'agir d'un adjectif, d'un verbe ou d'un nom.

Les expériences nous ont amené à confirmer que les phrases-clés en langue arabe sont généralement sous la forme de phrases nominales et leur longueur ne dépasse pas trois mots. Par conséquent, dans notre système proposé, un mot composé est sélectionné comme candidat ***phrase-clé***, si sa longueur est comprise entre 1 et 3 mots et vérifier l'un les modèles linguistiques suivants :

- Nom
- Nom + (nom ou adjectif).
- Nom + Verbe + Nom.
- Nom + Nom + Nom
- Nom+ Nom+ Verbe
- Nom+ Verbe+ Verbe

Pour l'analyse morphosyntaxique, nous avons utilisé le système d'analyse morphosyntaxique Alkhlil pour les textes arabes [67] mis en œuvre dans la plate-forme SAFAR. En effet, comme première étape, nous analysons la sortie d'Alkhlil qui génère une variation des formes grammaticales possible d'un syntagme. Si, l'une de ces formes grammaticales générées vérifie nos modèles linguistiques, le syntagme est sélectionné comme candidat de phrase-clé.

4.2 C-value pour sélectionner les phrases-clés pertinentes

Chaque mot-clés peut être généré avec un certain nombre de phrases clés auxquelles fait partie, de sorte que l'utilisation de la fréquence normalisée de l'occurrence pour mesurer la pertinence des expressions de mots clés devient inutile. En fait, les k-top phrases clés pertinentes comprendront également des sous-phrases-clés non pertinentes. Par conséquent, afin de surmonter ce problème, nous proposons d'utiliser la *C-value* comme mesure statistique afin de classer les phrases clés et de résoudre le problème de sous-phrases-clé. En fait, la *C-value* a été initialement proposée pour définir des expressions terminologiques potentielles et est basée sur la normalisation de la fréquence des mesures d'occurrence en tenant compte de la longueur des phrases-clés candidates et des phrases-clés apparaissant comme imbriquées dans des phrases clés plus longues. La *C-value* de la ***phrase-clés*** A est définie par [82]:

$$\mathrm{C}-vlaue(A)=\begin{cases}\dfrac{f(A)}{|A|}, \text{si } A \text{ est } non\ imbriqués\\ \dfrac{f(A)}{|A|}-\sum_{B\in Ta}\dfrac{f(B)}{|B|}, sinon\end{cases} \qquad \textbf{(5.4)}$$

où

- *f(A)* est la fréquence d'occurrence de la ***phrase-clés*** A dans le document.
- | *A* | Est la longueur de la phrase clés A.
- *Ta* est l'ensemble de toutes les phrases clés incluses dans A.

Et le score S (A) de la ***phrase-clé*** A devient alors :

$$S(A)=\begin{cases}C-value(A), \text{si } 3\geq|A|\geq 1\\ 0, sinon\end{cases} \qquad \textbf{(5.5)}$$

5 Expériences, résultats et discussion

5.1 Description des expériences

Pour mettre en évidence l'intérêt d'utiliser notre système proposé pour l'extraction des phrases-clés, nous présentons dans cette section des études expérimentales en utilisant des documents arabes où on compare les trois systèmes d'extraction de mots-clés arabes ***KP-Miner***, ***KpST*** et ***Improved KpST***. La comparaison se concentre sur la complexité des trois systèmes et la qualité de leurs phrases clés générées.

5.2 Expérience 1

Le but de l'expérience est d'évaluer la qualité des phrases clés extraites par les trois systèmes à l'aide d'un expert. Le tableau 25 présente les 7 principales phrases-clés extraites des 3 documents texte en langue arabe choisies au hasard parmi différents sujets. Comme nous l'observons dans ce tableau, les phrases-clés extraites par les trois systèmes sont pertinentes pour le sujet de document correspondant. Cependant, ceux extraites par notre système ***improved KpST*** sont plus respectueuses de la forme des phrases clés, ce qui n'est pas le cas pour les autres systèmes. Les phrases-clés générées par le ***KP-Miner*** peuvent être considérées comme des termes multi-mots. Et pour les phrases-clés générées par le système ***KpST***, nous pouvons observer que beaucoup d'entre elles (en caractères gras dans le tableau 20) doivent être ignorées parce qu'elles ne respectent pas les règles de la grammaire arabe et donc elles n'ont pas de sens.

Titre de Doc	**KP-Miner**	**KpST**	**Improved KpST**
الاقتصاد السعودي	- الميزانية العامة الجديدة - ريال سعودي - الميزانية العامة - الاقتصاد السعودي - سامبا - المالي السابق - المملكة العربية السعودية	- المملكة العربية السعودية - تأسيس البنك السعودي - نشاط البنك السعودي - البنوك التجارية السعودية - الخدمات البنكية الإسلامية - **العربية السعودية يشار** - الشركة السعودية المحدودة	- البنك السعودي - التجارية السعودية - المملكة العربية السعودية - تأسيس البنك السعودي - نشاط البنك السعودي - السوق السعودية - الخدمات البنكية الإسلامية
مستقبل البحث العلمي في العراق	- البحث العلمي - مراكز البحوث - الصناعي - العراق - الوطنية - مستقبل الأبحاث - الجامعات	- فلسفة البحث العلمي - مستقبل البحث العلمي - أخلاقية البحث العلمي - **البحث العلمي تأسس** - البحث العلمي والتقني - **البحث العلمي تبدأ** - مركز بحوث وحمايتها	- البحث - فلسفة البحث العلمي - مستقبل البحث العلمي - أخلاقية البحث العلمي - البحث العلمي والتقني - الأبحاث العلمية - مركز بحوث
ما يجب أن تعرفه عن الحول	- عضلات العين - حركة العينين - عين - مريض - حالة - سن - العين السليمة	- المريض العين السليمة - وعلاج كسل العين - **العين يبدأ الطبيب** - عضلات العين الخارجية - إحدى عضلات العين - العين المصابة الداخل - ودرجة كسل العين	- العين الخارجية - العين أكثر - العين محدودة - تحريك العين - بياض العين - باستخراج العين - كسل العين

Tableau 25 : *Evaluation subjective*

5.3 Expérience 2

Le but de cette étude expérimentale est de mesurer la performance de phrases-clés extraites pour les trois systèmes ***KP-Miner***, ***KpST*** et ***Improved KpST***, lorsque ces dernières font partie d'une autre application de fouille de texte. À cette fin, nous intégrons les trois systèmes avec le classificateur naïve Bayesien pour la classification des documents. La figure 47 décrit notre étude expérimentale.

Corpus : l'utilisation de 7 catégories sélectionnées à partir de corpus de test (Corpus of Contemporary Arabic (CCA))[22]. Ce dernier contient des documents provenant de sites Web et de radio Qatar. Le tableau 26 présente un sommaire de ce corpus.

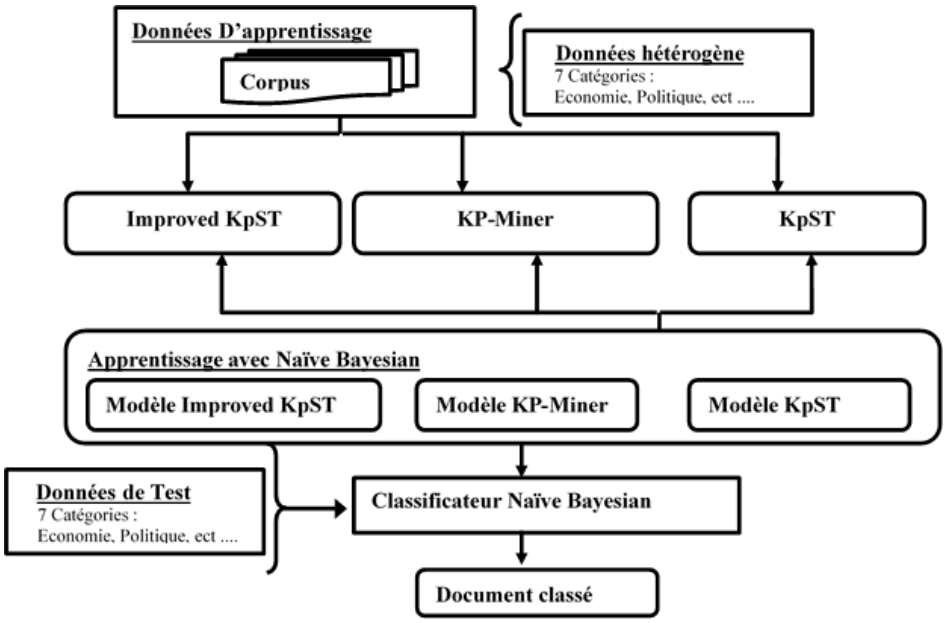


Figure 47 : *Description de l'étude expérimentale* [81]

Catégories	Nombre of Documents	Nombre of Termes
Economie	29	67 478
Education	10	25 574
Sante	32	40 480
Politique	9	46 291
Religion	19	111 199
Science	45	104 795
Interviews	24	58 408
Tourisme	61	46 093

Tableau 26 : *Corpus of Contemporary Arabic (CCA)*

Le corpus a été divisé aléatoirement en deux parties : une partie pour l'apprentissage et l'autre pour les tests.

- Les documents d'apprentissage présentent 70% des documents par catégorie.
- Les documents de test, d'autre part, représentent 30% des documents de chaque catégorie.

[22] http://www.comp.leeds.ac.uk/eric/latifa/research.htm

Nous avons comparé les trois systèmes de catégorisation en termes de F1-Mesure pour les trois systèmes d'extraction de phrases-clés sur le même corpus. La figure 48 présente les résultats de mesures F1 obtenus. La mesure F1 est calculée en utilisant les mesures de rappel et de précision comme suit :

$$F1 - measure = (2 * Précision * Rappel)/(Précision + Rappel) \quad \textbf{(5.6)}$$

- La précision : le nombre de documents pertinents retrouvés rapporté au nombre de documents total proposé par le moteur de recherche pour une requête donnée.
- Le rappel : le nombre de documents pertinents retrouvés au regard du nombre de documents pertinents que possède la base de données.

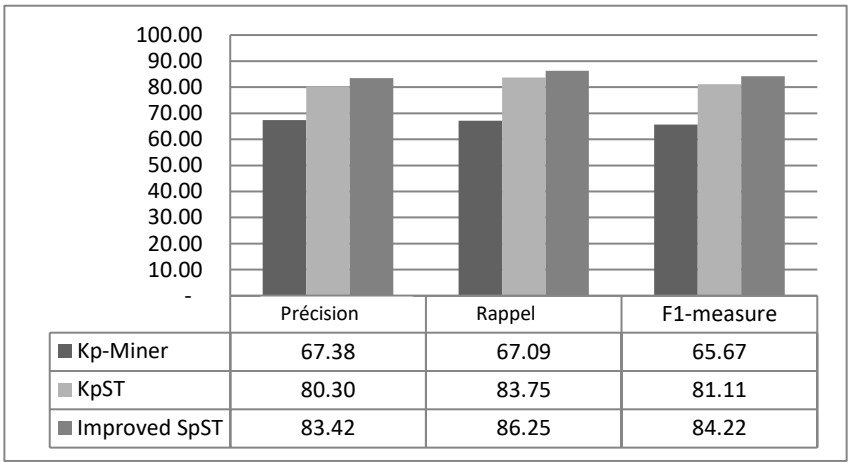
Mathématiquement, la précision et le rappel est définie comme suit :

$$Précision = TP/(TP + FP) \quad \textbf{(5.7)}$$

$$Rappel = TP/(TP + FN) \quad \textbf{(5.8)}$$

où

- *TP : Vrai positif,* le nombre de documents pertinents retrouvés.
- *TN : Vrai négatif,* le nombre de documents non pertinents et retrouvés.
- *FP : Faux positif,* le nombre de documents pertinents et non retrouvés
- *FN : Faux négatif,* le nombre de documents non pertinents et non retrouvés.



	Précision	Rappel	F1-measure
Kp-Miner	67.38	67.09	65.67
KpST	80.30	83.75	81.11
Improved SpST	83.42	86.25	84.22

Figure 48 : *Résultats de l'étude expérimentale*

Cette étude expérimentale montre que notre système ***Improved KpST*** améliore largement et très significativement les performances de catégorisation avec près de 16% par rapport au système ***KP-Miner***.

6 Conclusion

La représentation de texte dans n'importe quelle application de fouille de texte peut être affectée négativement par trop d'informations bruyantes dans les documents et produit des résultats insatisfaisants. Comme les phrases clés résument très concisément le document en éliminant le bruit et en sélectionnant les phrases les plus pertinentes, elles peuvent être utilisées comme une nouvelle représentation pour les documents.

Dans ce chapitre, nous avons présenté notre système Improved KpST pour l'extraction de phrases-clés en arabe. Les améliorations comprennent l'ajout d'un filtre linguistique et l'utilisation d'une nouvelle mesure de la pertinence des phrases clés basée sur la mesure C-value.

L'efficacité de notre système proposé est appuyée par des études expérimentales utilisant des documents arabes et comparant le système Improved KpST à KP-Miner et KpST. La comparaison relative à la qualité des phrases-clés générées, montre que notre système Improved KpST améliore largement les performances de catégorisation avec près de 16% par rapport au système ***KP-Miner***.

CHAPITRE 6. Nouvelle méthode d’indexation basée sur les phrases-clés

1 Introduction

La Recherche d'Information (RI) n'est pas un domaine récent, il date des années 40. Une des premières définitions de la RI a été donnée par Salton : « la recherche d'information est un domaine qui a pour objectif, la représentation, l'analyse, l'organisation, le stockage et l'accès à l'information » [1].

Plusieurs tâches se regroupent sous le vocable de la RI. La plus ancienne est la recherche documentaire, on y trouve également d'autres tâches plus au moins récentes comme le filtrage d'information, l'extraction d'information, la recherche d'information multilingue, les questions réponses et la recherche d'information sur le web.

Le processus d'indexation se considère comme une étape hors-ligne primordiale dans n'importe quel système de recherche d'information, qui peut influencer d'une façon positive ou négative la performance de résultat d'un tel système.

Dans ce chapitre, nous présentons le processus d'indexation. Nous commençons par citer l'ensemble des problèmes liés à l'utilisation d'indexation basée sur les mots clés simples dans les SRI en particulier pour la langue arabe, ainsi que les solutions présentées dans la littérature pour surmonter ces problèmes. Par la suite, nous présentons notre SRI dédié à la langue arabe basé sur le concept des phrases-clés.

2 Le processus d'indexation

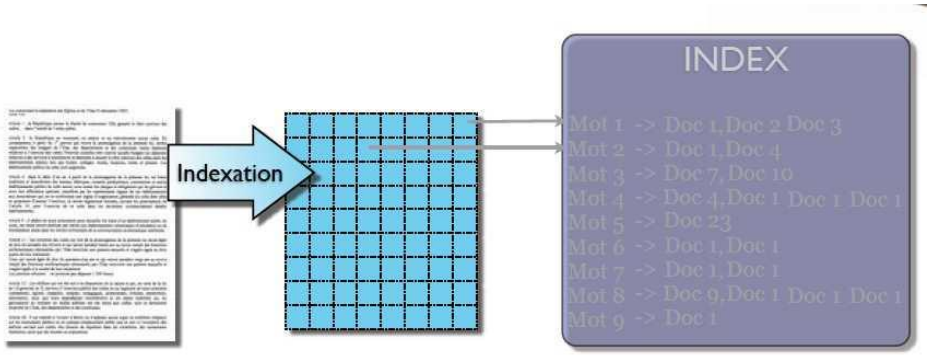


Figure 49 : *Processus d'indexation [83]*

Pour que la recherche d'information se réalise avec des coûts acceptables, il convient d'effectuer une opération fondamentale sur les documents de la collection. Cette opération est nommée indexation [83]. Elle consiste à associer à chaque document une liste de mots-clés appelés aussi descripteurs, susceptibles de représenter au mieux le contenu des documents.

Le principe de stockage des informations dans l'index est celui d'un annuaire inversé (Figure 49). Chaque terme est associé à une liste de documents y contenant une occurrence. L'index stocke les informations sur les sources dans un format pivot. La richesse de ce format pivot, qui structure l'index, va être garante de la qualité de recherche.

La finalité de l'indexation est donc de produire une représentation synthétique des documents, formée de termes peuvant être extraits de trois manières :

- ***Manuelle*** : chaque document de la collection est analysé par un spécialiste du domaine ou un documentaliste. L'indexation manuelle assure une meilleure précision dans les documents restitués par le SRI en réponse aux requêtes des utilisateurs [84]. Néanmoins, cette indexation présente un certain nombre d'inconvénients liés notamment à l'effort et au prix qu'elle exige (en temps et en nombres de personnes). De plus, cette indexation est subjective, car elle est liée au facteur humain. En effet, différents spécialistes peuvent indexer un document avec des termes différents. Il se peut même qu'un spécialiste indexe différemment un document, à différents moments.
- ***Semi-automatique :*** la tâche d'indexation est réalisée ici conjointement par un programme informatique et un spécialiste du domaine [85]. Le choix final des descripteurs revient à l'indexeur humain. Dans ce type d'indexation, un langage d'indexation contrôlé est généralement utilisé.
- ***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 racinisation), 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 enfin la création de l'index et éventuellement sa compression. Nous détaillons ces différentes étapes dans les sous sections suivantes.

2.1 L'analyse lexicale

Elle permet de convertir un texte de document en une liste de termes. Un terme est un groupe de caractères constituant un mot significatif [86]. L'analyse lexicale permet de reconnaître les espaces de séparation des mots, les chiffres, les ponctuations, etc.

2.1.1 Elimination des mots vides

Les mots vides (article, proposition, conjonction, etc.) sont des mots non significatifs dans un document, car ils ne traitent pas le sujet du document.

On distingue deux techniques pour éliminer les mots vides :

- L'utilisation d'une liste préétablie de mots vides (aussi appelée *anti- dictionnaire* ou *stop-list*),

- L'élimination des mots ayant une fréquence qui dépasse un certain seuil dans la collection.

L'élimination des mots vides réduit la taille de l'index, ce qui améliore le temps de réponse du système. Cependant, elle peut réduire le taux de rappel en réponse à des requêtes bien spécifiques (par exemple, la requête *be or not to be*).

2.1.2 Normalisation

La normalisation consiste à représenter les différentes variantes d'un terme par un format unique appelé lemme ou racine. Ce qui a pour effet de réduire la taille de l'index. Plusieurs stratégies de normalisation sont utilisées : la table de correspondance, l'élimination des affixes (l'algorithme de Porter [44] pour les langue latine), la troncature, l'utilisation des N-grammes [87]. L'inconvénient majeur de cette opération est qu'elle supprime dans certains cas la sémantique des termes originaux, c'est le cas par exemple des termes derivate/derive, activate/active, normalisés par l'algorithme de Porter.

2.1.3 Choix des descripteurs

Le choix des descripteurs consiste à déterminer le type d'unités élémentaires pour représenter les documents. L'objectif est d'avoir une représentation des documents permettant une moindre perte d'information sémantique possible. Dans la littérature, nous distinguons plusieurs types de descripteurs [88] présentés ci-dessous :

- **Les mots simples** : ce sont les mots dans un document en éliminant les mots vides,
- **Les lemmes** ou **les racines** des mots extraits.
- **Les N-grammes** : qui sont une représentation originale d'un texte en séquence de N mots consécutifs. On trouve des utilisations de bi-grammes et trigrammes dans la recherche d'information.
- **Les mots composés :** groupes de mots ou expression (phrase en anglais) sont souvent plus riches sémantiquement que les mots qui les composent pris séparément. Par exemple, le mot composé "imprimante laser" est plus précis que "imprimante" et "laser" pris isolément. Cet argument a conduit à leur large utilisation en RI.
- **Phrase-clé :** une phrase ou une partie d'une phrase qui contient une séquence de mots qui exprime le sens et le but de tout paragraphe donné.
- **Les concepts :** qui sont des expressions pris généralement d'une structure conceptuelle, telle que les thésaurus ou les ontologies.

2.1.4 Création de l'index

Au terme du processus d'indexation, un ensemble de structures de données sont créées. Ces dernières permettent un accès efficace à la représentation des documents. Le fichier inverse est la

structure de données la plus utilisée [83]. Il enregistre pour chaque descripteur les identificateurs des documents qui le contiennent et sa fréquence dans chacun de ces documents.

Généralement, les structures de données sont compressées avant d'être enregistrées sur le disque, ce qui permet de réduire la taille de l'index. Parmi les méthodes de compression utilisées on peut citer la méthode Elias Gamma [89] qui opère au niveau bit requérant ainsi beaucoup d'opérations pour la compression et la décompression. D'autres méthodes plus efficaces, opérant au niveau octet ont été proposées dans [90]. D'autres caractéristiques sur un document, permettant de calculer la pertinence a priori d'un document indépendamment de toute requête, peuvent être calculées et stockées a ce stade [91].

2.2 Au-delà des mots simples

La majorité des approches (modèles) développées en RI se basent sur l'utilisation des mots simples comme unités de représentation des documents et des requêtes, souvent appelées représentation en sac de mots. Ces approches posent deux problèmes, l'ambiguïté des mots et leur disparité [92].

- ***L'ambiguïté des mots,*** se rapporte à des mots lexicalement identiques et portant des sens différents. Ce problème conduit à avoir des documents non pertinents en réponse à une requête contenant des mots ambigus. Par exemple, des documents sur la « programmation java » peuvent être renvoyés en réponse à la requête « aéroport de java », car le terme java contient plus d'un sens (ile, programmation, etc.) [92].
- ***La disparité des mots (en anglais word mismatch),*** se réfère à des mots lexicalement différents mais portant un même sens. Ce problème implique que des documents pertinents ne sont pas retrouvés en réponse à une requête, car ils utilisent des mots différents que ceux de la requête pour exprimer le même concept. Par exemple, des documents contenant le terme «***tablette tactile*** » peuvent ne pas être retrouvés en réponse à une requête « ***I-pad***» [92].

Diverses approches ont été proposées pour remédier à ces problèmes. Ces approches permettent d'incorporer ou d'utiliser des informations conceptuelles ou sémantiques dans les méthodologies de recherche. Nous présentons ci-dessous les quatre types d'approches les plus utilisées :

1. Indexation sémantique
2. Indexation conceptuelle
3. Indexation par des mots composés
4. Indexation par des phrases-clés

2.3 Indexation sémantique

L'indexation sémantique consiste à représenter les documents et les requêtes par les sens des termes qu'ils contiennent plutôt que par les termes eux-mêmes. Ce type d'indexation se base sur les techniques de désambiguïsation sémantique (Word Sense Disambiguation WSD).

La désambiguïsation sémantique est une tâche qui a pour but la sélection du sens approprié pour un terme dans un contexte donné [93]. En d'autre terme, faire disparaître l'ambiguïté d'un mot en ne retenant qu'un seul sens.

L'étude de Krovetz *et al* [94] a été la première et la plus élaborée à s'être penchée sur la pertinence de la relation de correspondance du sens des termes dans la requête et les documents. Ils ont découvert que lorsqu'une requête est bien formulée et décrit bien le besoin en information, alors l'ambiguïté est moindre. Considérant les deux requêtes suivantes : « big bank » et « bank economic financial monetary fiscal », alors la deuxième requête est moins ambiguë que la première. Ceci est dû à ce qu'ils ont appelé l'effet de collocation : des termes isolement ambigus contribuent ensemble à désambiguïser implicitement le sens des termes (bank dans l'exemple).

Ils ont aussi étudié la distribution des sens d'un terme dans les documents. Ils ont trouvé que certains sens du terme occurrent plus fréquemment que les autres.

Voorhees [95] a proposé une méthode de désambiguïsation basée sur Wordnet, qui est une structure conceptuelle organisée autour de la notion de synset. Un synset regroupe des termes (simples ou composés) ayant un même sens dans un contexte donné. Les synsets sont liés par différentes relations telles que l'hyperonymie (is-a) et son inverse, l'hyponymie (part-of) [88].

Pour déterminer le sens d'un mot ambigu, les synsets (sens) de ce mot sont classés en utilisant la valeur de cooccurrence calculée entre le contexte de ce mot et un voisinage (Voorhees l'a appelé hood) contenant les mots du synset dans la hiérarchie de WordNet. Le synset le mieux classé est alors choisi comme le sens approprié du mot ambigu analysé.

Voorhees a expérimenté cette approche sur une collection de tests désambiguïsés (les requêtes de la collection de tests sont aussi désambiguïsées manuellement) par rapport aux performances du même processus sur la même collection dans son état d'origine (ambigu). Les tests ont été effectués sur les collections CACM [1]s, CISI, Cranfield 1400, MEDLINE, et Time [96]. Les résultats obtenus sur chacune de ces collections montrent une nette dégradation des performances du SRI. Ceci est expliqué probablement par le taux faible de désambiguïsation.

Sanderson [97] a étudié l'effet de la désambiguïsation sémantique en RI, particulièrement l'effet d'une désambiguïsation incorrecte sur les performances de la RI. Il a mené un ensemble d'expérimentations en utilisant la collection Reuters. Il a constaté que l'introduction de l'ambiguïté artificielle dans la collection Reuters ne dégrade pas les performances du système. Il a aussi constaté que les requêtes courtes, comprenant un à deux termes sont fortement affectées par l'introduction

de l'ambiguïté, alors que les requêtes longues le sont beaucoup moins (ce qui confirme l'effet de la collocation noté par Krovetz).

Par contre, le travail inverse désambiguïsation de la collection ambiguë de Reuters, affecte les performances du système, et cela selon le taux de performance de la désambiguïsation effectué (qui est contrôlable). Avec un taux de performance de désambiguïsation égal à 75%, il a constaté que les performances du système se dégradent. Il a conclu que la désambiguïsation est utile lorsque le taux de performance de la désambiguïsation est supérieur à 90%.

Comme Sanderson, Gonzalo *et al* [98] ont mesuré l'effet de la désambiguïsation erronée sur les performances de la RI. Ils ont utilisé trois schémas d'indexation : les termes, le sens des termes (pas de prise en compte de la synonymie) et les synsets de WordNet. Des expérimentations ont été effectuées sur la collection SEMCOR**,** une partie du corpus Brown. Cette collection (SEMCOR) est désambiguïsée avec les synsets de WordNet.

Leur expérimentation est effectuée en deux étapes. La première consiste à évaluer l'effet de la désambiguïsation (l'effet de l'indexation avec le sens du terme et le synset). La seconde étape consiste à évaluer l'effet de la désambiguïsation incorrecte comme l'a fait Sanderson.

Les résultats obtenus dans la première partie montrent qu'un gain de 11% de performance est obtenu avec l'utilisation des sens des termes, et un gain de 14 % avec l'utilisation des synsets de WordNet.

Quant aux résultats obtenus dans la seconde étape ; ils montrent qu'avec un taux d'erreur de désambiguïsation inférieur à 90% les performances se dégradent, ce qui rejoint le constat fait par Sanderson. En revanche, avec une indexation par synset, les performances du système restent meilleures que celles de la configuration basique et cela avec un taux d'erreur de désambiguïsation de 30%. Pour un taux d'erreur de désambiguïsation variant entre 30% et 60%, les résultats ne montrent pas de différence significative avec l'indexation par mots-clés (configuration basique). Ce qui est en désaccord avec le constat de Sanderson.

En résumé, l'application des techniques de désambigüisation sémantique en RI présente des limitations en terme de calcul et d'efficacité. Des résultats mixtes ont été reportés dans de récentes séries d'expérimentations, réalisées à CLEF 2008 et à CLEF 2009, dans la tâche Robust–WSD [99].

2.4 Indexation conceptuelle

L'indexation conceptuelle consiste à représenter un document par un ensemble de concepts. Ces concepts sont tirés de structures conceptuelles, qui peuvent être génériques (cas de WordNet pour la langue anglaise) ou spécifique à un domaine (cas de MESH pour le domaine médical). Les structures conceptuelles peuvent être construites manuellement, automatiquement ou semi-automatiquement. Ces structures incluent les taxonomies de concepts, les ontologies, les réseaux sémantiques, les

dictionnaires, les thésaurus, etc., et qui diffèrent dans la forme de représentation utilisée et dans les relations entre concepts considérées.

Plusieurs travaux en RI ont utilisé ce type d'indexation dans des domaines spécifiques comme le domaine du sport [100], le domaine légal [101], ou dans un domaine générique [88], [102].

Woods [102] a présenté une méthode d'indexation conceptuelle, qui consiste à extraire automatiquement les descripteurs conceptuels (concepts) à partir des documents sans faire référence à aucune ressource, puis à organiser ces concepts de manière dynamique sous forme d'une taxonomie de concepts.

L'index conceptuel obtenu peut servir alors à la recherche ou à la consultation dans la collection des documents. L'organisation des concepts dans la taxonomie se base sur la relation de subsumption (is-a), dans laquelle chaque concept identifié est relié à ses concepts parents. Pendant la phase de la recherche, l'algorithme de recherche permet de déterminer l'emplacement de la requête dans la taxonomie.

La méthode ainsi décrite, a été évaluée sur la collection de pages du manuel d'UNIX et sur d'autres collections de Sun Microsystems. Les résultats obtenus ont montré que cette méthode d'indexation améliore le taux de succès de 13% par rapport à la meilleure stratégie d'indexation classique (*TWIDF* : Term Weighted Inverse Document Frequency). Le taux de succès est une mesure qui est définie comme le pourcentage de requêtes pour lesquelles une bonne réponse est obtenue dans les dix premiers documents retournés.

Baziz *et al.* [88] ont défini une méthode d'indexation conceptuelle basée sur l'utilisation de l'ontologie linguistique WordNet. Chaque document est représenté sous forme d'un réseau sémantique particulier (appelé noyau sémantique), dans lequel les nœuds représentent les concepts et les arcs (bidirectionnels) représentent la distance sémantique entre concepts liés.

Après expérimentations, les résultats obtenus ont montré que cette méthode d'indexation conceptuelle n'améliore pas les résultats obtenus avec une indexation classique (mots-clés). Par contre, les résultats obtenus avec une combinaison des deux types d'indexation (classique et conceptuelle) ont montré une nette amélioration de la précision.

2.5 Indexation par des mots composés

L'indexation par des mots composés est une technique qui permet l'utilisation des mots composés comme unités d'indexation. Ceci a pour objectif une représentation plus précise du contenu sémantique des documents et des requêtes. L'idée d'utiliser les mots composés comme unités d'indexation est que ces derniers sont moins ambigus et plus précis que les mots simples. Par exemples : le terme « *java* » est ambigu, par contre les mots composés « *ile de java* » et « *langage java* » sont non ambigus ; le terme « *voiture électrique* » est plus spécifique que l'un des deux termes « *voiture* » et « *électrique* ».

L’intuition est claire, les mots composés aident à construire des unités d’indexation non ambiguës et plus précises et peuvent par conséquent améliorer la précision de la RI. Cinq paramètres sont généralement à considérer dans l’exploitation des mots composés comme unités d’indexation.

- **La directionnalité** : c'est-à-dire l’ordre des termes. Dans certains cas, la préservation de l’ordre est importante pour préserver le sens de l’unité d’indexation. Par exemple, « Recherche d’information », dans d’autre cas l’ordre n’est pas important, « Recherche et développement ». Peu de travaux existent en RI où sont utilisés les mots composés directionnels (ex : [103]). La plupart des travaux exploitant les mots composés sont basés sur la non directionnalité de ces derniers [104] [105]. Cependant, Fagan *et al* [106] ont rapporté des problèmes dans l’utilisation des mots composés non directionnels.
- **La distance** : l’intensité de liens entre termes opérationnalisée à travers la distance entre les termes formant le mot composé (l’adjacence ou la non-adjacence des termes). La distance reflète la proximité sémantique entre termes. La capture de cette proximité est importante pour la recherche d’information. Les études effectuées en RI sur l'extraction des mots composés supposent que la cooccurrence des mots dans les éléments fortement structurés (c.-à-d., une phrase) est plus significative que dans les éléments moins structurés (c.-à-d., des paragraphes ou des sections). Ainsi, la recherche sur l'extraction des mots composés a été dominée par l'analyse de phrase. *Martin et al.* [107] ont constaté que 98% des combinaisons syntaxiques associent les termes qui sont dans la même phrase et sont séparés par cinq mots au plus. *Fagan et al.* [106] a constaté que la restriction de l'extraction des mots composés à une fenêtre de distance de cinq termes est presque aussi efficace que des mots composés extraits dans une phrase sans une telle restriction, soutenant ainsi les résultats de Martin *et al* [107]. D'autres travaux [108] [109] ont adopté cette hypothèse et ils ont utilisé une fenêtre de cinq mots pour l’extraction des mots composés.
- **La taille des mots composés** : un mot composé peut être de n’importe quelle longueur supérieure ou égale à 2. Dans la pratique les mots composés longs conduisent à des index très spécifiques qui sont généralement moins utiles pour la RI.
- **La pondération des mots composés** : les différents schémas de pondération proposés pour l’attribution d’un poids à un mot simple dans un document, prennent généralement en considération trois facteurs : le facteur de pondération local (*tf*), qui mesure l’importance du terme dans le document ; un facteur de pondération globale, mesurant la représentativité globale du terme dans la collection (*idf*) et un facteur de normalisation qui prend en compte la longueur du document.

  Cependant, pour les mots composés, il n’y pas de schéma de pondération bien accepté. En général, trois approches sont proposées pour la pondération des mots composés :

- ✓ L'utilisation de la fréquence (*tf*) du mot composé dans le document [110]; en se basant sur le fait que la fréquence d'un terme est corrélée avec son importance [111].
- ✓ L'adaptation de schéma de pondération (*tf x idf*) appliqué pour les mots simples. Comme c'est le cas dans [88].
- ✓ L'utilisation des mesures d'association, telle que l'information mutuelle [112].

- **Repérage des mots composés** : trois approches principales existent dans la littérature pour le repérage et l'extraction des mots composés.
  - ✓ **Approches linguistiques** : ces approches [113] se basent sur une analyse syntaxique partielle ou sur l'utilisation de patrons (templates) syntaxiques pour détecter les mots composés [114][115] [116]. Le plus souvent, un ensemble de patrons syntaxiques comme (NOM NOM) ou (NOM PREP NOM) est utilisé pour l'identification. Malgré les nombreuses études consacrées à ce problème, il n'existe pas encore, à notre connaissance, une méthode effective qui permet de distinguer les termes des non termes d'un point de vue syntaxique. Des exemples d'outils issus de ces approches sont TreeTagger [117]**,** AZ NOUN PHRASER de l'université de l'Arizona. Cependant, ces approches souffrent d'un inconvénient majeur puisque elles sont basées sur des règles, et ces règles sont dépendantes de la langue.
  - ✓ **Approches statistiques** : ces approches se basent sur la cooccurrence des termes dans le corpus pour extraire les mots composés [118][119] [120]**,** et cela en partant de l'hypothèse que des termes (souvent réduits à deux ou trois mots) qui apparaissent ensemble dans le texte sont susceptibles de représenter un concept. Les mots composés sont extraits ici soit en se basant sur leurs fréquences observées dans le corpus soit par l'utilisation des mesures d'association qui déterminent le degré d'association entre les mots composants.
    - ➢ **Les mesures d'association** : les mesures d'association permettent de calculer « un score d'association » pour chaque paire de termes candidat dans le corpus. Ce score indique le potentiel de ce candidat d'être reconnu comme un mot composé. Plusieurs mesures d'association ont été proposées dans la littérature, telles que l'information mutuelle et le coefficient de Dice [121]. Toutes ces métriques adoptent le postulat suivant : « les mots composés sont ceux dont les composants apparaissent ensembles plus souvent que par hasard », cela est obtenu en comparant la fréquence observée dans le corpus et la fréquence attendue (qui se base sur l'hypothèse d'indépendance des termes). Pour la reconnaissance de mots composés de taille supérieure à deux (taille>2), des algorithmes ont été proposés [122], des mesures ont été définies [123] et des extensions des métriques précédentes ont été proposées [124].

- ✓ **Approches mixtes** : ces approches se basent sur les régularités statistiques et les patrons syntaxiques pour l'extraction des mots composés [125] [126][127][128]. Fagan [104], [106] a comparé l'apport pour la RI des mots composés extraits statistiquement et des mots composés extraits linguistiquement, en utilisant l'analyse syntaxique, la troncature et la normalisation.

L'évaluation a montré que les mots composés extraits linguistiquement ont donné des résultats semblables ou plus faibles que les résultats obtenus avec les mots composés extraits statistiquement. Les gains de performance constatés en utilisant les mots composés extraits statistiquement dans son expérience étaient de l'ordre de 17% à 39%.

Les approches statistiques ont un avantage considérable puisqu'elles ne nécessitent aucune autre information ou ressource pour l'extraction des mots composés. Elles exploitent seulement les informations apparaissant dans le corpus, d'où leurs flexibilité et portabilité (i.e. elles ne dépendent ni de la langue du corpus ni du domaine traité par le corpus).

2.6 Indexation par des phrases-clés

L'indexation par des phrases clés est un cas particulier de l'indexation par des mots composés. Généralement, nous pouvons définir la ***phrase-clé*** comme des mots composés respectant la forme grammaticale d'une phrase. Les phrases clés ont été utilisées dans plusieurs applications de fouille de texte tel que le résumé automatique de document [129]. Dans le cadre de cette thèse nous proposons d'utiliser les phrases-clés comme nouvelle méthode d'indexation différente de l'indexation par les mots composés.

3 Notre Système proposé pour la recherche d'information basée sur les phrases-clés

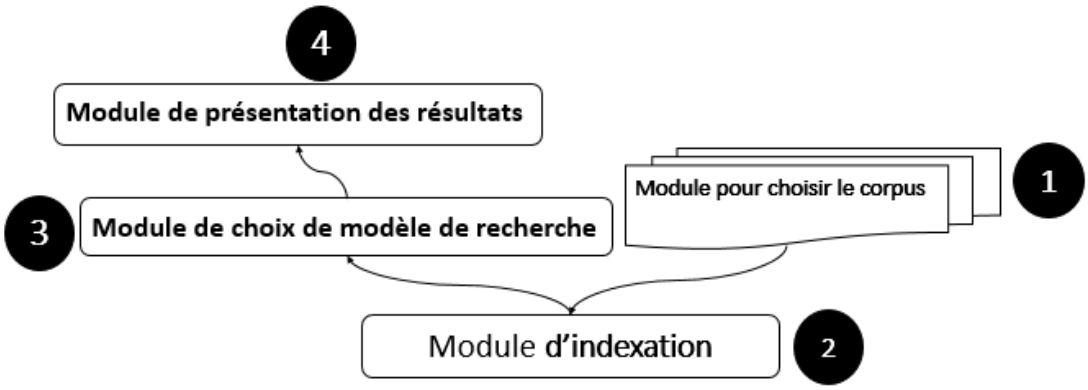


Figure 50 : *Nouvelle Système d'indexation basée sur les Phrases-Clés*

Le système de recherche d'information dédié à la langue arabe que nous avons proposé qui se base sur les phrases-clés, se compose (Figure 50):

Module pour choisir le corpus : une interface permettant à l'utilisateur de choisir le corpus pour la recherche (Figure 51).

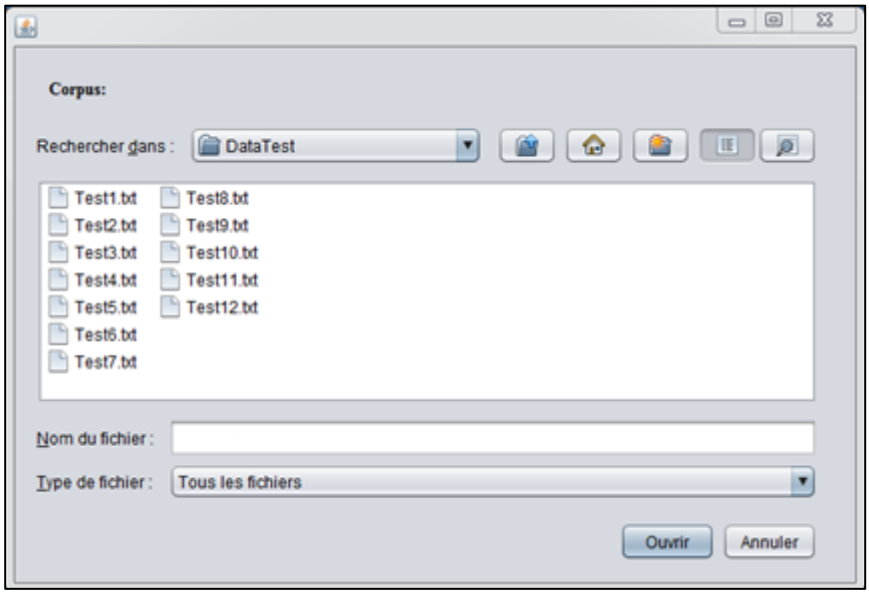


Figure 51 : *Interface pour choisir le corpus*

Module d'indexation : une interface permettant à l'utilisateur de choisir le type d'index. Comme illustré à la figure 52. L'utilisateur a le choix entre une indexation par des mots clés ou une indexation à base de phrases clés en utilisant soit le système *KP_Miner* [77] soit notre système *Improved KpST*.

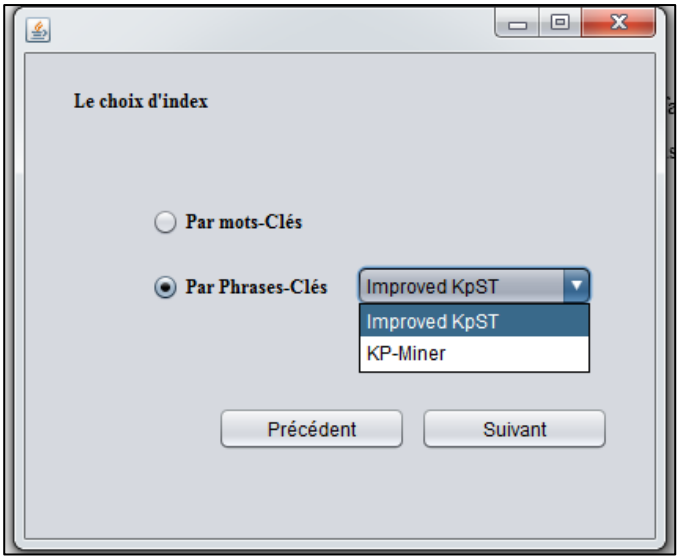


Figure 52 : *Interface pour choisir le modèle d'indexation*

Module de choix de modèle de recherche : une interface pour le choix du modèle de recherche d'information, comme illustré à la figure 53, l'utilisateur peut choisir entre trois modèles.

- Smart Boolean

- Modèle de l'espace vectoriel
- Modèle probabiliste en utilisant la méthode BM25

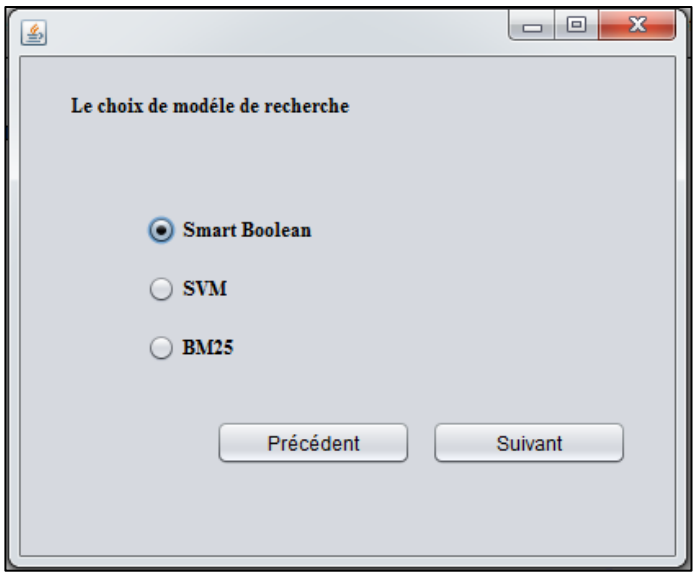


Figure 53 : *Module pour choisir le modèle de recherche*

Module de présentation des résultats : pour ce module nous utilisons le système de regroupement thématique de résultats de recherche que nous avons déjà détaillé dans les chapitres 3 et 4.

4 Evaluation de notre Système SRI : Enjeux et défis

Afin de mettre en évidence l'intérêt de notre proposition d'indexation à base des Phrases-Clés, nous avons intégré notre nouvelle méthode d'indexation dans la plate-forme OpenSource Terier[23]. Cette dernière est dédiée à la recherche d'information, elle implémente les différents modules intervenant dans le processus de RI classique et offre en plus un cadre pour l'évaluation des résultats de recherche d'information pour un système donné. Le choix de cette plate-forme est dû aussi à sa capacité à traiter de grandes collections de documents telles que les collections TREC[24].

Par ailleurs, nous signalons que l'étape d'évaluation permette de valider l'intérêt de n'importe quelle nouvelle méthode d'indexation proposée dans le cadre d'un système de recherche d'information, cette validation est basée principalement sur deux mesures Rappel et Précision. En effet, la précision donne le pourcentage de réponses correctes, tandis que le rappel donne le pourcentage des réponses correctes qui sont retournées par le système. Notons qu'en pratique, nous cherchons un bon compromis entre le rappel et la précision. Afin d'évaluer un système de RI, nous utilisons une courbe qui trace le rappel par rapport à la précision (ou vice versa) pour un corpus de test et d'évaluation.

[23] http://terrier.org/
[24] http://trec.nist.gov/data/test_coll.html

Notons également que notre nouvelle méthode d'extraction des Phrase-Clés a été déjà testée et validée avec succès dans le chapitre précèdent en l'intégrant dans un système de catégorisation des documents en langue arabe.

De plus nous avons mis en place notre système d'indexation et recherche d'information dédié à la langue arabe, cependant nous n'avons pas pu réaliser des expériences d'évaluations à cause de manque de ressource langagière en langue arabe. En effet, les collections de tests existantes pour la langue arabe d'une part sont très rares et d'autre part ne sont pas libres.

5 Conclusion

Dans ce chapitre nous avons passé en revue les principaux concepts de processus d'indexation dans les SRI. Nous avons, particulièrement, proposé un nouveau système d'indexation et de recherche d'information dédié à la langue arabe basé sur les phrases-clés. Vu que la majorité de collections de tests existants ne sont pas libres, nous avons intégré notre système d'extraction de phrases-clés dans un système de classification pour valider notre contribution. Les détails d'expérimentation sont présentés dans le chapitre 5.

Conclusion de la partie 3

La représentation de texte dans n'importe quelle application de fouille de texte peut être affectée négativement par trop d'informations bruyantes dans les documents et produire des résultats insatisfaisants. Comme les phrases clés résument très concisément le document en éliminant le bruit et en sélectionnant les phrases les plus pertinentes, elles peuvent être utilisées comme une nouvelle représentation pour les documents.

Dans cette partie, nous avons présenté les contributions concernant l'indexation. Nous avons commencé par notre système Improved KpST pour l'extraction de phrases-clés en arabe dans le chapitre 5. L'efficacité de notre système proposé est prouvée par des études expérimentales en utilisant des documents arabes où nous avons comparé les trois systèmes Improved KpST, KP-Miner et KpST. La comparaison montre que notre système Improved KpST améliore largement et très significativement les performances de catégorisation par rapport au système KP-Miner de près de 16%.

Dans le chapitre 6, nous avons passé en revue les principaux concepts de processus d'indexation dans les SRI. Nous avons développé un nouveau système d'indexation et de recherche d'information dédié à la langue arabe basé sur les phrases-clés. Vu que la majorité de collections de tests existants ne sont pas libres, nous avons intégré notre système d'extraction de phrases-clés dans un système de classification pour valider notre contribution.

Conclusion et Perspectives

Récemment, la langue arabe est devenue l'une des langues les plus utilisées dans le Web. Cependant, la majorité des solutions existantes pour améliorer l'utilisation du Web ne tiennent pas compte des caractéristiques de cette langue. Ce travail de thèse consiste à développer un système moderne de recherche d'information dédié aux documents web Arabes.

Cette thèse s'inscrit dans le cadre d'un projet qui vise à créer et améliorer les différentes composantes d'un *Système d'Indexation et Recherche d'Information pour la langue arabe,* dans le but de remédier aux différents problèmes menés par la complexité de cette langue dans le domaine de *Text Mining*. Dans notre travail, nous avons pu recenser et catégoriser l'ensemble des problèmes liés d'une part au processus de consultation des résultats de recherche web et d'autre part au processus d'indexation des documents. Notons que dans le cadre de la consultation, les moteurs de recherche existants tel que Google, Yahoo, Bing retournent une liste ordonnée d'une dizaine de milliers de snippets (métas-données), les utilisateurs ne consultent que les premières pages, et par conséquent les documents situés à la fin de la liste ne sont que très rarement consultables bien qu'ils puissent être pertinents. Au niveau du processus d'indexation, la méthode d'indexation basée sur les mots-clés pose un problème d'ambiguïté, ce qui influence négativement les résultats des systèmes de recherche d'information pour les différentes langues en particulier la langue arabe.

Objectifs atteints

Les différentes contributions proposées pour soutenir le domaine de *Recherche d'Information* pour la langue arabe dans le cadre de notre thèse, sont résumées comme suit :

1. Proposition d'un système interactif basé sur l'algorithme STC, permettant le regroupement des résultats de recherche pour les utilisateurs arabes.

2. Proposition d'un système basé sur le FCA (Formal Concept Analysis) qui permet un regroupement conceptuel et fournit une interface de consultation hiérarchique sur deux niveaux.

3. Proposition d'un système nommé KpST qui permet l'extraction de phrases clés, basé sur l'algorithme d'arbre de suffixes.

4. Amélioration du système KpST, en ajoutant une couche de filtrage linguistique, et l'utilisation d'une nouvelle mesure pour le calcul de score basée sur C-Value, ce système est nommé *improved-KpST*.

5. L’utilisation du système *improved-KpST*, a montré à travers les expériences menées, l’amélioration de la qualité de clustering et de catégorisation en comparaison avec la représentation textuelle complète des documents arabes.

Perspectives

Les différentes voies explorées dans le cadre de cette thèse débouchent sur plusieurs perspectives. Nous présentons ci-dessous celles qui nous paraissent les plus prometteuses pour l’amélioration des différentes composantes du *Système d’Indexation et Recherche d’Information pour la langue arabe* :

- Comme perspective, vu le problème d’absence d’un corpus de test disponible gratuitement aux chercheurs dans le domaine de la recherche d’information dédié à la langue arabe, nous visons collaborer avec des experts linguistiques pour construire un corpus de test qui sera disponible gratuitement et répond parfaitement aux besoins des chercheurs de ce domaine.

- Comme deuxième perspective, nous visons à paramétrer notre SRI afin d’intégrer des ontologies de domaines (**الأحاديث النبوية؛ الاقتصاد؛الرياضة....**) dont l’objectif est de construire un moteur de recherche personnalisé pour un domaine spécifique.

Bibliographie

[1] G. Salton And M. J. Mcgill, *Introduction To Modern Information Retrieval*. New York: Mcgraw-Hill International, 1983.

[2] M. H. Haddad, “Extraction Et Impact Des Connaissances Sur Les Performances Des Systèmes De Recherche D’information,” *Phd Thesis*, Joseph Fourier, 2002.

[3] T. Strzalkowski, *Natural Language Processing In Large-Scale Text Retrieval Tasks*. In *Text Retrieval Conference (Trec-1)*, 1992.

[4] A. F. Smeaton, “Progress In The Application Of Natural Language Processing To Information Retrieval Tasks,” *Comput. J.*, Vol. 35, No. 3, PP. 268–278, 1992.

[5] F. Ataa Allah, “Information Retrieval: Applications To English And Arabic Documents,” *Phd Thesis*, Université De Mohamed V, 2008.

[6] B. Piwowarski, “Technique D’apprentissage Pour Le Traitement D’informations Structurées: Application A La Recherche D’information,” *Phd Thesis*, Université De Paris 6, 2003.

[7] A. Spoerri, “Infocrystal: A Visual Tool For Information Retrieval,” In *Visualization ’93*, 1993, PP. 150–157.

[8] F. W. Lancaster And A. J. Warner, *Information Retrieval Today*. Information Resources Press Arlington, USA, 1993.

[9] R. S. Marcus, “Intelligent Assistance For Document Retrieval Based On Contextual, Structural, Interactive Boolean Models,” In *Proceedings Of Riao94 Conference On Intelligent Multimedia Systems And Management*, 1994, PP. 27–43.

[10] E. A. Fox, “Extending The Boolean And Vector Space Models Of Information Retrieval With P-Norm Queries And Multiple Concept Types,” Université De Cornell, 1983.

[11] E. Fox And S. Sharat, *A Comparison Of Two Methods For Soft Boolean Interpretation In Information Retrieval*. Technical Report Tr-86-1, Virginia Tech, Departement Of Computer Science, 1986.

[12] K. Bharat And M. R. Henzinger, “Improved Algorithms For Topic Distillation In A Hyperlinked Environment,” *21st Acm Sigir Conf. Res. Dev. Inf. Retr.*, 1998.

[13] D. Weiss, “A Clustering Interface For Web Search Results In Polish And English,” *Master Thesis*, Poznan University Of Technology, Poland, 2001.

[14] O. Zamir And O. Etzioni, “Grouper: A Dynamic Clustering Interface To Web Search Results,” *Comput. Networks*, Vol. 31, No. 11–16, PP. 1361–1374, 1999.

[15] C. Carpineto, S. Osiński, G. Romano, And D. Weiss, “A Survey Of Web Clustering Engines,” *Acm Comput. Surv.*, Vol. 41, No. 3, PP. 1–38, 2009.

[16] D. Zhang And Y. Dong, “Semantic, Hierarchical, Online Clustering Of Web Search Results,” *Adv. Web Technol. Appl.*, PP. 69–78, 2004.

[17] P. Ferragina And A. Gull, “The Anatomy of a Hierarchical Clustering Engine for web-page, News and Book Snippets,” *Fourth IEEE International Conference on Data Mining*, 2004.

[18] S. Osinski, J. Stefanowski, And D. Weiss, “Lingo : Search Results Clustering Algorithm Based On Singular Value Decomposition,” *Intelligent Information Processing and Web Mining. Advances in Soft Computing*, vol 25, PP. 359–368, 2004.

[19] G. Mecca, S. Raunich, A. Pappalardo, And D. Santoro, “Noodles : A Clusteri<ng Engine For The Web,” *Comput. Networks*, PP. 496–500, 2007.

[20] D. R. Tobergte And S. Curtis, “Indexing By Latent Semantic Analysis,” *J. Chem. Inf. Model.*, Vol. 53, No. 9, PP. 1689–1699, 2013.

[21] D. R. Cutting, K. R. David, P. O. Jan, And T. W. John, “Scatter / Gather : A Cluster-Based Approach To Browsing Large Document Collections,” *15th Ann Int’l Sigir’92/Denmark-6/92*, 1992.

[22] D. Boley, M. Gini, R. Gross, E. H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, And J. Moore, "Partitioning-Based Clustering For Web Document Categorization," *Decis. Support Syst.*, Vol. 27, No. 3, PP. 329–341, 1999.

[23] I. S. Dhillon, "Co-Clustering Documents And Words Using Bipartite Spectral Graph Partitioning," *Proc 7th Acm Sigkdd Conf*, PP. 269–274, 2001.

[24] M. E. J. Newman, "Finding Community Structure In Networks Using The Eigenvectors Of Matrices," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, Vol. 74, No. 3, 2006.

[25] P. Chen, H. Xie, S. Maslov, And S. Redner, "Finding Scienti C Gems With Google," *Condens. Matter Phys.*, No. 1, PP. 1–6, 2003.

[26] C. Carpineto And G. Romano, "Exploiting The Potential Of Concept Lattices For Information Retrieval With Credo," *J. Univers. Comput. Sci.*, Vol. 10, No. 8, PP. 985–1013, 2004.

[27] S. Brin And L. Page, "The Anatomy Of A Large Scale Hypertextual Web Search Engine," *Comput. Networks Isdn Syst.*, Vol. 30, No. 1/7, PP. 107–17, 1998.

[28] C. Neuhaus, E. Neuhaus, And A. Asher, "Google Scholar Goes To School: The Presence Of Google Scholar On College And University Web Sites," *J. Acad. Librariansh.*, Vol. 34, No. 1, PP. 39–51, 2008.

[29] R. Abbès And M. Boualem, "Dissymétrie Entre L'indexation Des Documents Et Le Traitement Des Requêtes Pour La Recherche D'information En Langue arabe," In *Taln 2008*, 2008.

[30] R. Abbès And J. Dichy, "Extraction Automatique De Fréquences Lexicales En Arabe Et Analyse D'un Corpus Journalistique Avec Le Logiciel Araconc Et La Base De Connaissances Diinar . 1," In *9es Journées Internationales D'analyse Statistique Des Données Textuelles*, 2008, PP. 31–44.

[31] M. Boualem, "Hahooa Arabic Web Directory & Natural Hahooa Arabic Web Directory Natural Language Processing For Arabic Information Retrieval," In *Acl Workshop On Arabic Language Processing*, 2001.

[32] M. H. Maâloul, "Approche Hybride Pour Le Résumé Automatique De Textes. Application A La Langue arabe," *Phd Thesis*, Université De Provence - Aix-Marseille I, 2012.

[33] S. 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," *Phd Thesis*, Université De Nantes, 2008.

[34] F. Chaar, "Traitement Automatique De La Langue arabe : La Modération En Arabe," Institut National Des Langues Et Civilisations Orientales (I.N.A.L.C.O), 2009.

[35] F. S. Douzidia, "Résumé Automatique De Texte Arabe," *Master Thesis*, Université De Montréal, 2004.

[36] S. Baloul, M. Alissali, M. Baudry, And M. Boula De, " Interface Syntaxe-Prosodie Dans Un Système De Synthèse De La Parole A Partir Du Texte En Arabe," In *Journées D'etudes Sur La Parole (Jep'02)*, 2002, PP. 329–332.

[37] K. S. Motaz, "The Impact Of Text Preprocessing And Term Weighting On Arabic Text Classification," *Master Thesis*, The Islamic University – Gaza, Palestine, 2010.

[38] L. H. Belguith, C. Aloulou, And A. Ben Hamadou, "Maspar : De La Segmentation A L'analyse Syntaxique De Textes Arabes," *Rev. Inf. Interact. Intell. I3*, Vol. 7, PP. 9–36, 2001.

[39] R. Ouersighni, "A Major Offshoot Of The Diinar-Mbc Project: Araparse, A Morphosyntactic Analyzer For Unvowelled Arabic Texts," In *Acl/Eacl 2001 Workshop On Arabic Language Processing*, 2001, PP. 9–16.

[40] J. Wu And Z. Wang, "Search Results Clustering In Chinese Context Based On A New Suffix Tree," *2008 Ieee 8th Int. Conf. Comput. Inf. Technol. Work.*, PP. 110–115, 2008.

[41] Y. Wang, W. Zuo, T. Peng, F. He, And H. Hu, "Clustering Web Search Results Based On Interactive Suffix Tree Algorithm," *2008 Third Int. Conf. Converg. Hybrid Inf. Technol.*, PP. 851–857, 2008.

[42] I. Sahmoudi And A. Lachkar, "Clustering Web Search Results For Effective Arabic Language Browsing," *Int. J. Nat. Lang. Comput.*, Vol. 2, No. 3, PP. 31–43, 2013.

[43] P. Weiner, T. R. Corporation, And S. Monica, "Linear Pattern Matching algorithms," *14th Annu. Ieee Symp. Switch. Autom. Theory*, PP. 1–11, 1973.

[44] M. F. Porter, “An Algorithm For Suffix Stripping,” *Program: Electronic Library And Information Systems*, Vol. 14, No. 3. PP. 130–137, 1980.

[45] S. Khoja And R. Garside, “Stemming Arabic Text,” *Technical Report,* Lancaster University, http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps, September 22, 1999.

[46] I. Sahmoudi And A. Lachkar, “Interactive System Based On Web Search Results Clustering For Arabic Query Reformulation,” In *2014 Third Ieee International Colloquium In Information Science And Technology (Cist)*, 2014, PP. 300–305.

[47] H. Froud, A. Lachkar, S. Ouatik, And R. Benslimane, “Stemming And Similarity Measures For Arabic Documents Clustering,” In *5th International Symposium On I/V Communications And Mobile Networks Isivc*, 2010.

[48] R. Wille, “Formal Concept Analysis As Mathematical Theory Of Concepts And Concept Hierarchies,” *Form. Concept Anal.*, PP. 1–33, 2005.

[49] Y. Zhang And B. Feng, “Clustering Search Results Based On Formal Concept Analysis,” *Information Technology Journal*. PP. 746–753, 2008.

[50] M. Chein, “Algorithme De Recherche Des Sous-Matrices Premières D’une Matrice,” *Bull. Mathématique La Société Des Sci. Mathématiques La République Social. Roum.*, Vol. 13, PP. 21–25, 1969.

[51] A. Guénoche, “Construction Du Treillis De Galois D’une Relation Binaire,” *Mathématiques Sci. Hum.,* Vol. 109, PP. 41–53, 1990.

[52] B. Ganter, “Ch1 & Ch2: Contexts, Concepts, And Concept Lattices,” *Form. Concept Anal. Methods Appl. Comput. Sci.*, 2003.

[53] J. Bordat, “Calcul Pratique Du Treillis De Galois D’une Correspondance,” *Mathématiques Sci. Hum. Math. Soc. Sci.*, Vol. 96, PP. 31–47, 1986.

[54] S. O. Kuznetsov, “A Fast Algorithm For Computing All Intersections Of Objects In A Finite Semi-Lattice,” *Autom. Doc. Math. Linguist.*, PP. 11–21, 1993.

[55] P. Becker, J. Hereth, And G. Stumme, “Toscanaj – An Open Source Tool For Qualitative Data Analysis,” *Adv. Form. Concept Anal. Knowl. Discov. Databases. Proc. Work. Fcakdd 15th Eur. Conf. Artif. Intell. (Ecai 2002)*, PP. 1–2, 2002.

[56] E. M. Norris, “An Algorithm For Computing The Maximal Rectangles In A Binary Relation,” *Rev. Roum. Mathématiques Pures Appliquées*, Vol. 23, No. 2, PP. 243–250, 1973.

[57] C. Carpineto And G. Romano, “A Lattice Conceptual Clustering System And Its Application To Browsing Retrieval,” *Mach. Learn.*, Vol. 24, No. 2, PP. 95–122, 1996.

[58] R. Godin, R. Missaoui, And H. Alaoui, “Incremental Concept Formation Algorithms Based On Galois (Concept) Lattices,” *Comput. Intell.*, Vol. 11, No. 2, PP. 246–267, 1995.

[59] P. Valtchev, D. Grosser, C. Roume, And M. R. Hacene, “Galicia: An Open Platform For Lattices,” In *Using Conceptual Structures: Contributions To 11th Intl. Conference On Conceptual Structures (Iccs 03)*, 2003, PP. 241–254.

[60] S. O. Kuznetsov And S. A. Obiedkov, “Comparing Performance Of Algorithms For Generating Concept Lattices,” *J. Exp. Theor. Artif. Intell.*, Vol. 14, No. 2–3, PP. 189–216, 2002.

[61] P. Burmeister, “Formal Concept Analysis With Conimp: Introduction To The Basic Features,” *Fachbereich Math. Tech. Univ. Darmstadt*, 2003.

[62] S. A. Yevtushenko, “System Of Data Analysis Concept Explorer,” In *Proceedings Of The 7th National Conference On Artificial Intelligence Kii*, 2000, PP. 127–134,.

[63] F. Vogt And R. Wille, “Toscana A Graphical Tool For Analyzing And Exploring Data,” In *Graph Drawing, Springer*, 1995, PP. 226–233.

[64] B. Koester, “Conceptual Knowledge Retrieval With Fooca: Improving Web Search Engine Results With Contexts And Concept Hierarchies,” *Lect. Notes Comput. Sci.*, PP. 176– 190, 2006.

[65] E. Nauer And Y. Toussaint, “Classification Dynamique Par Treillis De Concepts Pour La Recherche D’information Sur Le Web,” In *Conférence En Recherche D’information Et Applications*, 2008, PP. 71–86.

[66] I. Sahmoudi And A. Lachkar, “Formal Concept Analysis For Arabic Web Search Results Clustering,” *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 29, No. 2, PP. 196–203, 2017.

[67] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, And M. Ould Abdallahi Ould Bebah, M Shoul, “Alkhalil Morpho Sys1: A Morphosyntactic Analysis System For Arabic Texts,” *Proceedings Of Int. Arab Conf. Inf. Technol.*, PP. 1–6, 2010.

[68] B. Ganter, “Two Basic Algorithms In Concept Analysis,” *Proceedings Of International Conference on Formal Concept Analysis*, PP. 312–340, 2010.

[69] A. Strehl, “Relationship-Based Clustering And Cluster Ensembles For High- Dimensional Data Mining,” *Phd Thesis*,Texas University, 2002.

[70] F. Geraci, M. Pellegrini, M. Maggini, And F. Sebastiani, “Cluster Generation And Cluster Labelling For Web Snippets: A Fast And Accurate Hierarchical Solution,” *String Process. Inf. Retr.*, Vol. 13, No. 10, PP. 25–36, 2006.

[71] P. D. Turney, "Learning Algorithms For Keyphrase Extraction," *Information Retrieval*, Vol. 2, PP. 203-336, 2000.

[72] Z. Liu, C. Liang, M. Sun, "Topical Word Trigger Model For Keyphrase Extraction," *Proceedings Of Coling 2012: Technical Papers*, PP.1715–1730, 2012.

[73] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, And C. G. Nevill-Manning, "Domain-Specific Keyphrase Extraction," In *IJCAI*, PP. 668-673, 1999.

[74] Z. Liu, P. Li, Y. Zheng, And M. Sun, "Clustering To Find Exemplar Terms For Keyphrase Extraction," *Proceedings Of Emnlp*, PP. 257–266, 2009.

[75] T. A. El-Shishtawy And A. K. Al-Sammak, "Arabic Keyphrase Extraction Using Linguistic Knowledge And Machine Learning Techniques," *Proceedings Of The Second International Conference On Arabic Language Resources And Tools*, 2012.

[76] R. Duwairi, F. Berzou, And S. Mecheter, "Hierarchical Clustering For Keyphrase Extraction From Arabic Documents Based On Word Context," *Qatar Foundation Annual Research Forum Proceedings*, Vol. 2011, Csp6, 2011.

[77] S. R. El-Beltagy And A. Rafea, “Kp-Miner: A Keyphrase Extraction System For English And Arabic Documents,” *Inf. Syst.*, Vol. 34, No. 1, PP. 132–144, 2009.

[78] I. Sahmoudi, H. Froud, And A. Lachkar, “A New Keyphrases Extraction Method Based On Suffix Tree Data Structure For Arabic Documents Clustering,” *Int. J. Database Manag. Syst.*, Vol. 5, No. 6, PP. 17–33, 2013.

[79] H. Froud, I. Sahmoudi, And A. Lachkar, “An Efficient Approach To Improve Arabic Documents Clustering Based On A New K Eyphrases Extraction,” In *Computer Science & Information Technology*, 2013, PP. 243–256.

[80] I. Sahmoudi And A. Lachkar, “Performance Evaluation Of Text Based Keyphrases Representation For Arabic Text Mining Applications,” In *5th International Conference On Arabic Language Processing*, 2014, PP. 236–242.

[81] I. Sahmoudi And A. Lachkar, “Towards A Linguistic Patterns For Arabic Keyphrases Extraction,” In *2016 International Conference On Information Technology For Organizations Development (It4od)*, 2016, PP. 1–6.

[82] K. Frantzi, S. Ananiadou, And H. Mima, “Automatic Recognition Of Multi-Word Terms: The C-Value/Nc-Value Method,” *Int. J. Digit. Libr.*, Vol. 3, No. 2, PP. 115–130, 2000.

[83] D. Manning, P. Raghavan, And H. Schute, “Introduction To Information Retrieval,” In *Cambridge University Press*, 2008.

[84] F. Ren, L. Fan, And J. Y. Nie, “Saak Approach: How To Acquire Knowledge In An Actual Application System,” In *International Conference On Artificial Intelligence And Soft Computing, Honolulu*, 1999, PP. 136–140.

[85] C. Jacquemin, B. Daille, J. Royanté, And X. Polanco, “In Vitro Evaluation Of A Program For Machine-Aided Indexing,” *Inf. Process. Manag.*, Vol. 38, No. 6, PP. 765–792, 2002.

[86] C. Fox, “Lexical Analysis And Stoplists,” In *Information Retrieval*, 1992, PP. 102–130.

[87] A. Hulth, “Improved Automatic Keyword Extraction Given More Linguistic Knowledge,” *Emnlp’03 Proc. 2003 Conf. Empir. Methods Nat. Lang. Process.*, No. 2000, PP. 216–223, 2003.

[88] M. Baziz, “Indexation Conceptuelle Guidée Par Ontologie Pour La Recherche D’information,” *Phd Thesis*,

Université De Paul Sabatier, 2005.

[89] I. Witten, A. Moffat, And T. Bell, "Managing Gigabytes: Compressing And Indexing Documents And Images," In *Van Nostrand Reinhold, New York*, 1994.

[90] H. Williams And J. Zobel, "Compressing Integers For Fast File Access," *Comput. J.*, Vol. 42, PP. 193–201, 1999.

[91] A. Das And A. Jain, "Indexing The World Wide Web: The Journey So Far," In *Next Generation Search Engines: Advanced Models for Information Retrieval* , IGI-Global, 2012, PP. 1–28.

[92] F. Boubekeur, "Contribution A La Définition De Modèles Flexibles De Recherche D'information Basés Sur Les Cp- Nets," *Phd Thesis,* Université Paul Sabatier, 2008.

[93] R. Navigli, "Word Sense Disambiguation: A Survey," *Acm Comput. Surv.*, Vol. 41, No. 2, PP. 1–69, 2009.

[94] R. Krovetz And W. B. Croft, "Lexical Ambiguity And Information Retrieval," *Acm Trans. Inf. Syst.*, Vol. 10, No. 1, 1992.

[95] E. M. Voorhees, "Using Wordnet To Disambiguate Word Senses For Text Retrieval," In *International Conference On Research And Development In Information Retrieval,* 1993, PP. 171–180.

[96] F. Song And W. B. Croft, "A General Language Model For Information Retrieval.," In *Proceedings Of International Acm Sigir Conference On Research And Development In Information Retrieval,* 1999, PP. 279–280.

[97] M. Sanderson, "Word Sense Disambiguation And Information Retrieval," In *Proceedings Of The 17th Annual International Acm-Sigir Conference On Research And Development In Information Retrieval, Springer- Verlag,* 1994, PP. 142–151.

[98] J. Gonzalo, F. Verdejo, I. Chugur, And J. Cigarrán, "Indexing With Wordnet Synsets Can Improve Text Retrieval," In *Proceedings The Coling/Acl Workshop On Usage Of Wordnet For Natural Language Processing,* 1998.

[99] E. Agirre, G. M. Di Nunzio, T. Mandl, And A. Otegi, "Clef 2009 Ad Hoc Track Overview: Robust–Wsd Task," In *Proceedings Of Clef. Springer,* 2009.

[100] R. K. Latifur, "Ontology-Based Information Selection," *Phd Thesis*, Faculty Of The Graduate School, University Of Southern California, 2000.

[101] K. Sparck Jones And C. J. Van Rijsbergen, "A Test For The Separation Of Relevant And Non-Relevant Documents In Experimental Retrieval Collections," *J. Doc.*, Vol. 32, No. 1, PP. 59–75, 1976.

[102] W. A. Woods, "Conceptual Indexing: A Better Way To Organize Knowledge.," *Tech. Rep. Smli Tr-97-61, Sun Microsystems Lab. Mt. View, Ca,* 1997.

[103] E. Mittendorf, B. Mateev, And P. Schäuble, "Using The Co-Occurrence Of Words For Retrieval Weighting," *Inf. Retr. Boston.*, Vol. 3, No. 3, PP. 243–251, 2000.

[104] J. L. Fagan, "The Effectiveness Of Non Syntactic Approach To Automatic Phrase Indexing For Document Retrieval," *J. Am. Sociely Inf. Sci.*, Vol. 40, No. 2, PP. 115–132, 1989.

[105] M. Mitra, C. Bucklcy, A. Singhal, And C. Cardie, "An Analysis Of Statistical And Syntactic Phrases," In *Proceedings Of The Fifth Conference On Computer Assisted Information Retrieval, Montreal, Canada,* PP. 200–214,1997.

[106] J. L. Fagan, "Experiments In Automatic Phrase Indexing For Document Retrieval:A Comparison Of Syntactic And Non-Syntactic Methods.," Cornell University, Ithaca, Ny, 1987.

[107] W. J. R. Martin, B. P. F. Ai, And P. J. G. Van Strenkenburg, "On The Processing Of Test Corpus: Froni Textual Data To Lexicographical Information," In *In Lexicography. Principles And Practice, R. R. K. I-Tartmann (Cd.), Academic Press, London*, 1983, PP. 77–87.

[108] D. Carmel, E. Amitay, M. Herscovici, Y. Maarek, Y. Petruschka, And A. Soffer, "Juru at TREC 10 - Experiments with Index Pruning," In *Proceedings Of The Text Retrieval Conference*, PP. 500–250, 2001.

[109] Y. Maarek, D. Berry, And G. Kaiser, "An Information Retrieval Approach For Automatically Constructing Software Libraries," *Ieee Trans. Softw. Eng.*, Vol. 17, No. 8, PP. 800–813, 1991.

[110] C. Khoo, S. Myaeng, And R. Oddy, "Using Cause-Effect Relations In Text To Improve Information Retrieval Precision.," *Inf. Process. Manag.*, Vol. 37, PP. 119–145, 2001.

[111] H. P. A. Luhn, “A Business Intelligence System.,” *Ibm J. Res. Dev.*, Vol. 2, No. 4, PP. 314–319, 1958.

[112] B. T. Bartell, G. W. Cottrell, And R. K. Belew, “Automatic Combination Of Multiple Ranked Retrieval Systems,” In *Proceedings Of The Acm Sigir Conference On Research And Development In Information Retrieval*, 1994, PP. 173– 181.

[113] P. Sheridan And A. F. Smeaton, “The Application Of Morpho-Syntactic Language Processing To Effective Phrase Matching,” *Inf. Process. Manag.*, Vol. 28, No. 3, PP. 349–370, 1992.

[114] N. Aussenac-Gilles, B. Biébow, And N. Szulman, “Revisiting Ontology Design: A Method Based On Corpus Analysis,” In *Proceedings Of The 12th International Conference On Knowledge Engineering And Knowledge Management*, 2000, PP. 172–188.

[115] D. Bourigault, “A Natural Language Processing Tool For Terminology Extraction,” In *Proceedings Of Euralex’96, Göteborg University, Department Of Swedish*, 1996, PP. 771–779.

[116] P. Church, K. And Hanks, “Word Association Norms Mutual Information And Lexicography.,” In *Proceedings Of The 28th Annual Meeting Of The Association For Computational Linguistics*, 1990, PP. 76–83.

[117] H. Schmid, “Probabilistic Part-Of-Speech Tagging Using Decision Trees.,” In *Proceedings Of International Conference On New Methods In Language Processing*, 1998, PP. 25–36.

[118] S. E. Robertson, “The Probability Ranking Principle In IR,” *J. Doc.*, Vol. 33, No. 4, PP. 294–304, 1977.

[119] G. Salton, “A Comparison Between Manual And Automatic Indexing Methods,” *J. Am. Doc.*, Vol. 20, No. 1, PP. 61–71, 1971.

[120] G. Sabah And B. Grau, “Compréhension Automatique De Textes,” In *Chap. 13, Ingéniérie Des Langues, Sous La Direction De J.M.Pierrel, Hermes*, 2000, PP. 293–307.

[121] D. Manning And H. Schütze, “Foundations Of Statistical Natural Language Processing,” *Mit Press*, 2000.

[122] A. Thanopoulos, N. Fakotakis, And G. Kokkinakis, “Identification Of Multiwords As Preprocessing For Automatic Extraction Of Lexical Similarities,” In *In 6th International Conference Text, Speech And Dialogue*, 2003, PP. 98–105.

[123] G. Dias, S. Guillore, J. Bassano, And J. G. P. Lopes, “Extraction Automatique D’unités Complexes: Un Enjeu Fondamental Pour La Recherche Documentaire,” *Trait. Autom. Des Langues*, Vol. 41, No. 2, PP. 447–472, 2000.

[124] B. T. Mcinnes, “Extending The Log-Likelihood Measure To Improve Collocation Identification,” University Of Minnesota, 2004.

[125] S. Boulaknadel, B. Daille, And A. Driss, “Multi-Word Term Indexing For Arabic Document Retrieval,” *Proc. - Ieee Symp. Comput. Commun.*, PP. 869–873, 2008.

[126] S. Liu, F. Liu, C. Yu, And W. Meng, “An Effective Approach To Document Retrieval Via Utilizing Wordnet And Recognizing Phrases,” In *Proceedings Of The 27th Annual International Conference On Research And Development In Information Retrieval. Acm Press, New York, Ny*, 2004, PP. 266–272.

[127] A. F. Smeaton And I. Quigley, “Experiments On Using Semantic Distances Between Words In Image Caption Retrieval,” In *Proceedings Of An International Acm Sigir Conference On Research And Development In Information Retrieval*, 1996, PP. 174–180.

[128] T. Strzalkowski, “Natural Language Information Retrieval,” In *Overview Of The Third Text Retrieval Confrerence (Trec3)*, 1995.

[129] P. Bhaskar, “Multi-Document Summarization Using Automatic Key-Phrase Extraction,” In *Proceedings Of The Student Research Workshop Associated With Ranlp 2013* , Hissar, Bulgaria, 9-11 September 2013, 2013, PP. 22–29.

[130] Y. Souteh, K. Bouzoubaa, "SAFAR platform and its morphological layer," *Eleventh Conference on Language Engineering ESOLEC’2011*, 14-15 December, Cairo, Egypt, 2011