République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur
et de la recherche scientifique

Université du 08 mai 45, Guelma

Faculté des sciences et de l'ingénierie
Institut de l'informatique

Mémoire présenté pour l'obtention
du diplôme de Magister en informatique
Option: Intelligence artificielle et Imagerie

Classification non supervisée de textes arabes appliquée à la recherche documentaire

Présenté par Kelaiaia Abdessalem

Devant le jury :

Président	: M.C	: H. Seridi	Université de Guelma
Rapporteur	: M.C	: H. F. Merouani	Université de Annaba
Examinateurs:	M.C	: Y. Tlili-Guiassa	Université de Annaba
	M.C	: L. Souissi-Meslati	Université de Annaba

Remerciements

الله سبحانه وتعالى أشكر على القوة التي أمدني بها حتى أكملت هذا العمل

Je remercie chaleureusement Monsieur Seridi Hamid, Maître de conférence à l'université 08 Mai 45 Guelma d'avoir accepté d'être le président de mon jury

Je n'oublierai pas de le remercier encore pour les grands efforts qu'il a fournis afin que notre formation aboutisse

Un remerciement particulier à Madame Yamina Tlili-Guiassa Maître de conférence à l'université Badji Mokhtar Annaba d'avoir accepté d'être l'examinatrice de ce mémoire ainsi que pour le grand intérêt qu'elle a porté à mon travail

Je remercie Madame Souissi-Meslati Labiba Maître de conférence à l'université Badji Mokhtar Annaba d'avoir accepté d'être l'examinatrice de ce mémoire

Ma reconnaissance va à Madame Hayet Farida Merouani, Mon encadreur, pour ses précieux conseils, ses encouragements et sa compréhension

Un très grand merci pour mon ami et collègue Karim Djelailia pour avoir été présent pour m'aider, me soutenir et m'encourager dans les moments les plus durs au cours de ces trois dernières années

Mes remerciements vont également:

aux enseignants de notre année théorique pour leurs encouragements et leurs compréhensions

à l'équipe COSI 2007 pour leur chaleureux accueil et spécialement Professeur Lakhdar Sais de l'université d'Artois et Professeur Nacira Benamrane de l'université d'Oran

aux Professeur Kareem Darwish de l'Université du Caire et Professeur Dunja MLADENIĆ de l'institut de J. Stefane, Slovénie pour nous avoir consacré un peu de leur temps

l'équipe : Leila, Lilia, Nassima et Nedjoua pour leur contribution à l'élaboration de ce travail (Questions et Documents pertinents)

A toute personne ayant contribué de près ou de loin à l'élaboration de ce travail,
Je dis merci.

Dédicace

Je dédie ce travail aux êtres qui me sont les plus chères au monde:

ma mère

mon frère Hamdi

mes sœurs Hakima, Samira, Lynda et Lilia

ces êtres qui ont été toujours à mes cotés pour me soutenir

et m'encourager

A la mémoire de mon père.

Résumé

La langue arabe a été et reste sujet de diverses recherches vu ses caractéristiques morphosyntaxiques. En effet, et mise à part l'orientation d'écriture qui est de droite à gauche, les deux principales caractéristiques de cette langue sont l'agglutination et la structure très particulière combinant schème et radical. Ces deux caractéristiques ont longtemps posé de problèmes dans le traitement automatique de cette langue.

Dans ce mémoire, nous avons appliqué une approche de classification non supervisée ou clustering sur une collection de textes en langue arabe, afin d'étudier la réaction de cette langue à un tel processus. Pour évaluer cette influence, nous avons fait recours à la recherche documentaire (RD). Une recherche documentaire classique emploie, généralement, des méthodes statistiques permettant le traitement des requêtes en langage naturel sur les corpus. Ces méthodes calculent la ressemblance entre la requête introduite et tous les documents du corpus pour fournir une liste ordonnée de documents. Malheureusement, les documents pertinents à la requête sont, généralement, mal positionnés voir inexistant sur cette liste, ce qui ne permet pas à l'utilisateur de les explorer.

Dans notre approche, avant d'effectuer une recherche documentaire, le corpus est soumis à une classification non supervisée, ensuite la liste des documents renvoyée est construite à partir des clusters formés selon le principe du plus proche représentant parmi les représentants des clusters par rapport à la requête introduite.

Plusieurs paramètres influents tels que le stemming (radicalisation), le nombre de clusters et la méthode de classification non supervisée sont étudiés.

Pour effectuer la classification nous avons choisi de tester deux méthodes, la première est la classification hiérarchique par agglomération, la deuxième est la méthode des k-médoïds.

Mots-clés: textes, langue arabe, classification non supervisée documentaire, clustering documentaire, classification hiérarchique, k-médoïds, stemming, TF-IDF, Recherche documentaire.

Abstract

The Arabic language has been and remains subject to various research morpho-syntactic due to its characteristics. Indeed, apart the orientation, which is a writing from right to left, the two main characteristics of this language are clumping and very particular structure combining pattern and radical. These two features have long been a problem in the automatic processing of the language.

In this work, we applied an approach to unsupervised classification (clustering) on a collection of texts in Arabic, in order to study the response of this language to such a process. To evaluate this influence, we have recourse to document retrieval (DR). Classical document retrieval employs generally statistical methods for the processing of queries in natural language on corpus. These methods calculate the similarity between the introduced query and all documents of the corpus to provide an ordered list of documents. Unfortunately, the relevant documents to this query are generally poorly positioned seen non existent on the list, which does not allow the user to explore it.

In our approach before making document retrieval, the corpus is submitted to an unsupervised classification process. After that, a hit list is built from clusters formed based on the nearest representative among the representatives of clusters in relation to the introduced query.

Several influential parameters such as stemming, the number of clusters and the method of unsupervised classification are studied.

To make the classification we have chosen to test two methods, the first is the agglomerative hierarchical clustering; the second is the method of k-medoïds.

Key-words: texts, Arabic language, document unsupervised classification, document clustering, hierarchical clustering, k-medoïds, stemming, TF-IDF, document retrieval.

ملخص

كانت ولا تزال اللغة العربية تخضع لمختلف البحوث و ذلك لخصائصها النحوية و الصرفية. فبغض النظر عن اتجاه الكتابة الذي هو من اليمين إلى اليسار، فإن من الخصائص الرئيسية لهذه اللغة اتصال الأحرف ببعضها البعض و التركيبة الجد خاصة التي تجمع بين شكل أو نمط الكلمة و جذرها. هاتان الخصيتان شكلتا منذ زمن طويل عائقا للمعالجة الأوتوماتكية لهذه اللغة.

في هذه المذكرة، قمنا بتطبيق تصنيف دون إشراف على مجموعة من النصوص باللغة العربية، و ذلك لدراسة مدى استجابة هذه اللغة لمثل هذه العملية. و لتقييم هذا التأثير، لجأنا إلى استعمال البحث الوثائقي أو المستندي الذي عادة ما يستخدم طرق إحصائية لمعالجة جملة بحث أو استفسار بلغة طبيعية أدخلت بغرض البحث عن مستندات داخل بنك مستندي. هذه الطرق الإحصائية تقوم بحساب التشابه بين الاستفسار و كل مستند من البنك المستندي حيث يتم استخلاص قائمة للمستندات. لسوء الحظ، فإن المستندات المناسبة لهذا الاستفسار عادة ما تكون سيئة الترتيب أو غير موجودة بتاتا على هذه القائمة، مما لا يسمح للمستخدم باستكشافها.

في دراستنا هذه و قبل إجراء البحث عن المستندات، يتم إخضاع البنك المستندي إلى عملية تصنيف، ثم يتم تشكيل قائمة المستندات التي سترجع انطلاقا من المصنفات أو المجموعات المشكلة و ذلك على أساس اقرب ممثل من بين ممثلي المجموعات إلى جملة البحث أو الاستفسار.

عدة عوامل مؤثرة قد تمت دراستها مثل طريقة ايجاد جذر الكلمة، عدد المجموعات أو المصنفات و الطريقة المستعملة لإيجادها.

لاختبار التصنيف اخترنا طريقتين، الأولى هي التصنيف التدريجي التجميعي و الثانية هي طريقة K-medoïds .

الكلمات الرئيسية : النصوص، اللغة العربية، التصنيف المستندي دون اشراف، التصنيف التدريجي، K-medoïds، ايجاد جذر الكلمة، TF-IDF، البحث الوثائقي أو المستندي.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Acronymes et Abréviations

AFP : Agence France Press

APT : Arabic part-of-speech tagger

ASCII : American Standard Code for Information Interchange

CCA : Corpus of Contemporary Arabic

CLARA : Clustering LARge Applications

CLARANS : Clustering Large Applications based on RANdomized Search

DEFT : Défi Fouille de Textes

MSA : Modern Standard Arabic

LDC : Linguistic Data Consortium

LSI : Latent Semantic Indexing

NIST : National Institute of Standards and Technology,

DR : Document Retrieval

SIAC : Segmentation et Indexation Automatiques de Corpus

PAM : Partitioning Around Médoïds

PERL : Practical Extraction and Report Language

TALN : Traitement Automatique du Langage Naturel

TFC : Term Frequency Cosine

TF-IDF : Term Frequency, Inverse Document Frequency

TREC : Text Retrieval Conference

UTF-8 : Unicode Text Format-8

XML : eXtensible Markup Language

YALE : Yet Another Learning Environment

Introduction

Il y a juste quelques années, les moteurs de recherche renvoyaient quelques pages Web pour une requête introduite. Dès lors, ce nombre a explosé depuis, des millions de pages apparaissent chaque jour, on se retrouve devant un océan d'informations électroniques sous plusieurs formes tirées de plusieurs sources telles que les agences de presse, les news groups, les courriers électroniques..., plus de 80% de ces informations sont stockées sous une forme textuelle [Witte, 2006]. Les utilisateurs qui sont les organismes commerciaux ou services, publics ou privés, les particuliers construisent à partir de cette source des bases documentaires gigantesques. Toute cette information serait sans intérêt si la capacité à y accéder efficacement n'est pas aussi fiable.

En effet, devant cette masse d'informations, la question d'accéder à l'information ne se posait plus, au contraire elle devenait : comment trouver l'information dont on a besoin, parmi toutes celles qui sont accessibles ?

Le développement des outils qui permettent la recherche d'une information particulière, l'exploration de toute une collection, l'analyse de l'information et plus, tout ça en un temps raisonnable est inévitable. De tels outils rentrent dans le cadre de la fouille de textes (*Text mining*).

Dans cette optique, plusieurs conférences sont périodiquement organisées, la plus connue est la conférence internationale de la recherche de texte *(Text REtrieval Conference,* TREC*).* Cette conférence est organisée annuellement aux *Etats-Unis* sous l'égide de l'institut national des standards et de la technologie *(National Institute of Standards and Technology,* NIST*)*, elle offre un forum d'évaluation et de discussions pour la communauté scientifique qui se consacre au traitement automatique des textes dans les différentes langues (y compris la langue arabe qui, quant à elle, a été sujet des recherches dès l'année 2001). Au cours de cette conférence, un ensemble de tâches différentes est proposé aux participants qui soumettent, à leurs tours, des résultats à autant de tâches qu'ils le souhaitent.

1. Classification automatique de textes

Un des axes de recherche de la conférence TREC et bien d'autres est la classification automatique de textes. En effet la recherche d'une information dans une base documentaire ou son exploration sont deux tâches très délicates, si cette dernière n'est pas convenablement

organisée. En faite, la performance d'un système de recherche documentaire est étroitement liée à une bonne organisation ou classification thématique de la source documentaire. On pourrait imaginer de demander à des humains de lire tous les textes dans la base documentaire et de les classer manuellement. Seulement cette tâche s'avère colossale si on est en face des centaines, voire des milliers de documents. Il apparaît alors très intéressant de pouvoir compter sur une application informatique qui, de façon automatique, assignerait ces textes à un ensemble prédéfini ou non de classes. C'est précisément là le but de la classification automatique de textes.

Les techniques de **la classification automatique de textes** sont répertoriées selon deux principales approches, la première est dite classification supervisée ou catégorisation et est basée sur l'apprentissage supervisé, la deuxième est dite classification non supervisée ou Clustering ou encore apprentissage non supervisée pour la classification[1], c'est dans cette dernière que notre étude se place.

L'apprentissage non supervisé pour la classification ou clustering, a déjà fait ses preuves dans le Data Mining et a cumulé un regain d'intérêt surtout avec l'augmentation de l'intérêt apporté à l'information textuelle et l'abondance des textes électroniques qu'une base documentaire peut contenir.

Deux approches de classification non supervisées sont généralement utilisées [Jain et al., 2000], [Turenne, 2000], [Jardino, 2004], la classification hiérarchique (ascendante ou descendante) qui permet de générer un arbre complet allant du regroupement de tous les éléments à classer dans une seule classe, à la racine de l'arbre, à la répartition de chaque élément dans sa propre classe, aux feuilles de l'arbre. L'autre méthode est le partitionnement direct en un nombre de classes spécifié à l'avance**.**

Quelque soit l'approche utilisée, un processus de classification automatique est précédé par une **phase de prétraitement** visant à normaliser la représentation des textes à classer. Divers outils tels que la tokenisation *(tokenization)*, la radicalisation *(stemming)*, la lemmatisation *(lemmatization)*, l'indexation *(indexing)*… sont utilisés dans le traitement des langues anglaise, française, allemande et autres langues européennes. Pour plusieurs de ces langues, des techniques de classification et de règles élaborées ont été testées pour la recherche de l'information et l'extraction de connaissances. Pour la langue arabe le chemin est encore long, quelques approches seulement fonctionnent. L'aspect **morphologique,** très

[1] Dans la littérature, on ne retrouve pas une distinction signifiante entre les trois appellations.

délicat de cette langue [Sawaf et al., 2001], qui n'apparaît pas dans les langues européennes, rend les techniques du prétraitement difficilement applicables.

2. Motivation

La nature de la langue arabe, le système d'écriture, l'orientation d'écriture, l'omission des voyelles et sa structure morphologique ont beaucoup ralentis les recherches sur cette langue dans le domaine de la classification automatique. Aussi, la difficulté de l'obtention d'un corpus en langue arabe, a obligé certains chercheurs a allé jusqu'à construire leurs propres corpus pour tester et valider leurs idées. Dans la littérature beaucoup de travaux se sont focalisés soit sur l'aspect morphologique de la langue arabe [Larkey et al., 2005] en développant des outils du prétraitement tel que le stemming et leur influence sur une recherche documentaire, soit pour la classification supervisée ou catégorisation. Par contre peu de recherches se sont intéressées à la tâche de clustering documentaire, ce qui nous a orienté vers cet axe de recherche.

3. Objectifs visés

Un des axes les plus importants du *Text mining* est la recherche documentaire *(DR : Document Retrieval),* celle ci représente l'extraction d'un ensemble de documents, jugés pertinents, à partir d'un corpus (une base documentaire ou encore une collection de textes), répondant à un besoin d'un utilisateur formulé à travers une requête en langage naturel. Un calcul d'indices de ressemblance entre la requête et chacun des documents de la collection cible est effectué. Suivant les valeurs de ces indices, une liste ordonnée de documents *(hit list)* est fournie à l'utilisateur. Cette liste est souvent si longue que les utilisateurs ne peuvent l'examiner entièrement, laissant ainsi de côté certains documents pertinents mal positionnés.

Dans les travaux de *Patrice Bellot* et *Marc El-bèze* [Bellot et El-bèze, 2000] une classification non supervisée appliquée à la liste des documents renvoyée par un système de recherche, améliore la précision d'une recherche documentaire. Ils ont aussi constaté l'influence du nombre de clusters utilisés sur la qualité de cette recherche. Cette étude s'est focalisée sur des corpus en langue anglaise et française.

Dans le présent travail, nous voulons tester la validité d'une approche similaire à celle de [Bellot et El-bèze, 2000] sur un corpus de textes en langue arabe, avec la différence que la classification non supervisée est appliquée à tout le corpus. Nous pouvons ainsi mesurer l'influence de la nature de la langue arabe sur la fiabilité d'une telle approche, d'une part sur des textes ayant subis des prétraitements, et d'autre part sur des textes à l'état brut *(raw text)*

et cela comparé à une recherche documentaire classique. Nos expérimentations sont menées sur plusieurs nombres de clusters, ce qui va nous permettre de mesurer non seulement la qualité de la classification obtenue mais aussi l'influence du nombre de clusters sur la précision d'une recherche documentaire.

4. Organisation du mémoire

Le présent mémoire présente deux parties distinctes.

Dans la première partie, constituée des trois premiers chapitres et intitulée état de l'art, nous commencerons par présenter une panoplie de techniques qui relèvent du prétraitement telles que la radicalisation (*stemming*), la lemmatisation (*lemmatization*), l'indexation et la représentation du texte, comme premier chapitre.

Le deuxième chapitre traite, en premier lieu, la classification automatique non supervisée ou clustering, d'ordre général, avec ses différentes approches et les techniques existantes. La deuxième partie de ce chapitre est consacrée à la classification non supervisée documentaire ainsi qu'à la recherche documentaire.

Le troisième chapitre est entièrement dédié à la langue arabe et ses particularités.

La deuxième partie est destinée à la classification non supervisée documentaire et son apport dans une recherche documentaire locale. Une méthodologie des recherches menées ainsi que l'approche proposée seront exposées. Les différentes expérimentations menées afin de concrétiser cette méthodologie sont décrites. Cette partie comprend le quatrième et le cinquième chapitres.

Le quatrième chapitre décrit les différentes phases de cette méthodologie, il commence par la phase de préparation des données, puis la phase de la classification non supervisée documentaire et enfin la phase qui décrit l'approche proposée.

Le cinquième chapitre donne une présentation, d'ordre général, des corpus de textes en langue arabe, puis une présentation du corpus utilisé, ensuite une description détaillée des moyens entrepris ainsi que les outils développés sera donnée. Nous enchaînerons, enfin, avec les différentes étapes des expérimentations menées suivies par plusieurs discussions.

Nous achèverons ce travail par une conclusion générale dans laquelle nous allons rappeler les résultats obtenus et nous évoquerons les perspectives immédiates à ce mémoire.

Chapitre I
Prétraitement,
indexation et représentation de textes

1. Introduction

La richesse du langage naturel, dans lequel est rédigée une grande masse d'informations, et sa large variabilité due essentiellement à la synonymie et à la polysémie, a causé de sérieux problèmes aux chercheurs dans le domaine du Traitement Automatique du Langage Naturel (TALN). Au début des recherches sur cet axe, l'approche syntaxique liée à la théorie des langages formels était la dominante par rapport aux autres approches numériques qui s'appuyaient sur les probabilités, elle a suivi la tradition linguistique en prenant la phrase comme unité fondamentale d'analyse et de traitement [Memmi, 2001]. L'analyse syntaxique de la phrase (en utilisant des grammaires formelles et des automates) a été le plus souvent considérée comme un préliminaire indispensable à l'interprétation sémantique.

Puis, une autre approche commençait à prendre de l'ampleur, elle est relativement indépendante du TALN syntaxique, mais plutôt liée aux statistiques. Elle découle des besoins de la classification et de la recherche documentaire. Ici, on cherche à calculer les probabilités de cooccurrences entre mots ou expressions, plutôt que de construire des structures syntaxiques,

Dans cette approche, quelque soit le traitement (classification, recherche documentaire), une préparation des textes s'impose. Elle consiste à transformer les textes, qui sont une suite de chaînes de caractères, en une représentation numérique facilement interprétable et manipulable par ce traitement, c'est le prétraitement. Dans ce qui suit nous allons décrire, en général, les étapes d'un prétraitement.

2. Prétraitement de texte *(text preprocessing)*

Le prétraitement de texte permet de formater les données textuelles et de les rendre directement exploitables par les traitements ultérieurs. Nous citons ici, les opérations les plus communément utilisées.

2.1. Tokenisation *(Tokenization)*

La tokenisation consiste à effectuer un nettoyage dans un texte [Baldi et al., 2003], en enlevant les expressions inutiles telles que les mots vides de sens, communément appelés *«stop words»*, les mots rares, les métadatas, les éléments formatés (exemple : les balises dans les documents XML). En suite, le texte est transformé en une liste de mots appelés *tokens*. Pour la langue arabe, deux tokeniseurs sont les plus connus, celui développé par *T. Buckwalter* [Buckwalter, 2002] et celui développé par *M. Diab* [Diab et al., 2004] intitulé *Diab tokenizer*.

Il faut noter que certains auteurs [Zhai, 2002] considèrent qu'une tokenisation est faite implicitement durant un traitement de stemming.

2.2. Désuffixation ou Radicalisation *(Stemming)*

Le stemming est une technique morphologique largement utilisée pour la préparation des textes dans une recherche documentaire [Korenius et al., 2004]. Elle consiste à rechercher la racine lexicale ou stem pour des mots en langue naturelle, et ceci, par l'élimination des affixes qui leurs sont rattachés, en d'autre terme regrouper sous un même identifiant des mots dont la racine est commune. Par exemple, en langue française les mots : déménageur, déménageurs, déménagement, déménagements, déménager, déménage, déménagera, etc. sont considérés comme des mots de la même racine «déménage». En langue arabe, les mots : حَامِلٌ , حَمْلٌ, مَحْمُولٌ , حَمَلَة sont des flexions du mot : «حَمَلَ ». Pour cela, des stemmers sont conçus, ils sont généralement destinés pour une langue bien spécifique sur laquelle une certaine expertise doit être élaborée. [Larkey et al., 2005] considère que l'utilisation d'un dictionnaire de stems (travaux de **[**Al-Kharashi et Evens, 1994**])** et l'analyse morphologique (travaux de **[**Buckwalter, 2002**])** sont une autre forme de stemming.

Plusieurs algorithmes de stemming ont été proposés; pour la langue anglaise, le plus connu est celui de *M. Porter* [Porter, 1980] intitulé *Porter's stemmer*[1], pour la langue arabe, on retrouve plusieurs stemmers, les plus connus sont : *Al-Stem*[2] **[**Darwish et Oard, 2002**]** développée par *K. Darwish* et *StemmerLight10* [Larkey et al., 2005] développé par *L. Larkley*, ce dernier est intégré dans le projet *Lemur* [Lemur, 2006].

[1] Sur le site http://www.comp.lancs.ac.uk/computing/research/stemming nous retrouvons une description détaillée de trois stemmers : *Porter'stemmer, Lovins'stemmer et Paice'Stemmer*

[2] *Al-Stem* a été modifié par *L.Larkey* pour qu'il puisse travailler avec l'encodage cp1256 (arabe Windows) http://www.microsoft.com/globaldev/reference/sbcs/1256.mspx

2.3. Lemmatisation *(Lemmatization)*

La lemmatisation consiste à retrouver l'entrée du dictionnaire pour une forme fléchie d'un mot, en d'autre terme, rechercher des lemmes en remplaçant les verbes par leur forme infinitive, les noms par leur forme au singulier et regrouper des mots dont la signification est la même alors même que leurs racines sont différentes en utilisant une analyse grammaticale. Par exemple, en langue française les mots «maison» et «baraque» ont le même sens [Dunoyer, 2004]. La lemmatisation est une tâche plus complexe que le stemming, elle repose, habituellement, sur l'utilisation de grandes bases de connaissances. Un algorithme nommé *TreeTagger* a été développé par *H. Schmid*, pour les langues anglaise, française, allemande et italienne [Schmid, 1994].

Pour la langue arabe, plusieurs lemmatiseurs ont vu le jour et seulement quelques uns ont été testés [Larkey et al., 2005]. Ces lemmatiseurs sont généralement intégrés dans des environnements d'analyses morphologiques (conjointement avec des étiqueteurs syntaxiques)[1], *Sebawai* [Darwish, 2002] est celui développé par *K. Darwish*, il est utilisé par les participants au cours de la conférence annuelle TREC. *T. Buckwalter* [Buckwalter, 2002] a développé un lemmatiseur intégré dans *Tim Buckwalter's morphological analyzer*, mais plutôt considéré comme étant un stemmer, il est disponible chez LDC (*Linguistic Data Consortium*) [LDC, 2007]. Un autre lemmatiseur développé par *M. Diab* [Diab et al., 2004] est incorporé dans un environnement d'analyse morphologique de la langue arabe.

Un autre outil intéressant est celui développé par *Ken Beesley et Tim Buckwalter*, *Xerox Arabic Morphological Analyser* [Xerox, 1997], qui est un système qui donne une analyse morphologique au mot suivi de la signification en langue anglaise.

Il faut noter que généralement un lemmatiseur est associer à un thésaurus qui, contrairement à un dictionnaire, ne donne pas d'informations relatives au sens et à l'emploi des mots, mais qui permet l'exploration à partir d'un concept (ou idée); les mots qui s'y rattachent et inversement.

Les étiqueteurs syntaxiques (*Part of Speech Tagger*) sont des outils d'analyse morphosyntaxique, qui peuvent être employés pour aider un processus de lemmatisation. Ces étiqueteurs consistent à donner une étiquette grammaticale (catégorie grammaticale) à un mot dans un texte brut. Plusieurs étiqueteurs ont été développés, le plus connu est celui de *E. Brill*

[1] Se reporter au site : http://www.irit.fr/~Mustapha.Baziz/Liens.htm pour plus de détails

[Brill, 1994]. Pour la langue arabe, nous citons celui de *S. Khoja* [Khoja, 2001] intitulé APT (*Arabic part-of-speech tagger*), ainsi que celui de *M. Diab* [Diab et al., 2004] intégré dans son environnement d'analyse morphologique

3. Indexation et représentation de documents

Les textes dans leurs états bruts ou non structurés ne sont pas directement interprétable par un classificateur, c'est pour cette raison qu'une procédure d'indexation est nécessaire [Sebastiani, 2002]. Cette procédure consiste à associer à chaque document du corpus un vecteur composé de termes, chaque terme est appelé terme d'indexation (ou attribut[1]). Le résultat, pour chaque document on aura un vecteur de termes d'indexation appelé aussi descripteur.

Formellement, si V est le vocabulaire contenant les termes qui apparaissent au moins une fois dans le corpus, un document dj est transformé en un vecteur :

$$dj = (w_{j1} , w_{j2} , ..., w_{jV}) \quad \textbf{(1.1)}$$

w_{jk} est appelé poids correspond à la contribution du terme qui a le rang k à la sémantique du document dj.

3.1. Choix des termes

Dans la littérature, on retrouve plusieurs approches pour la représentation du terme d'indexation.

3.1.1. Représentation en sac de mots (*Bag of words*)

C'est la représentation la plus simple, elle repose sur le principe de transformer les textes en vecteurs dont chaque composante représente un mot. Dans ce cas un mot peut être pris comme une séquence de caractères appartenant au vocabulaire, ou bien, de façon plus pratique, comme étant une séquence de caractères non-délimiteurs séparés par des caractères délimiteurs (caractères de ponctuation) [Gilli, 1988] [in Jalam, 2003]. Les composantes du vecteur sont une fonction de l'occurrence des mots dans le texte du document. Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots, d'où l'appellation de «**sac de mots**» *(bag of words)*.

3.1.2. Représentation par des phrases

La représentation en sac de mots présente le défaut majeur de ne pas grader la trace des mots dans le texte, ce qui l'a rendu peu informative. Cet inconvénient a inspiré certains

[1] Tout au long de ce mémoire nous ne faisons pas de distinction entre terme et attribut

auteurs tels que **[**Fuhr et Buckley, 1991**],** à utiliser les phrases comme unité de représentation. Une phrase est plus informative que les mots seuls, car elle a l'avantage de conserver l'information relative à la position du mot. Une telle représentation a permis de préserver les qualités sémantiques mais les qualités statistiques sont largement dégradées, le grand nombre de combinaisons possibles entraîne des fréquences faibles et trop aléatoires [Lewis, 1992].

Une autre idée, qui consiste à utiliser les phrases statistiques comme unités de représentation, a donné de bons résultats [Caropreso et al., 2001]. Une phrase statistique est un ensemble de mots contigus (mais pas nécessairement ordonnés) qui apparaissent ensembles mais qui ne respectent pas forcement les règles grammaticales.

Il est important de noter, qu'à notre avis, la représentation en phrases ne convient pas à la langue arabe vu la forme agglutinante de cette dernière. Néanmoins, un prétraitement pourrait bien donner de bons résultats, ce qui est d'ailleurs confirmé dans [Boulaknadel, 2005].

3.1.3. Représentation avec des racines lexicales (stems) et des lemmes

Dans le modèle de représentation en sac de mots, chaque flexion d'un mot est considérée comme un descripteur différent et donc une dimension de plus, les techniques de stemming et de lemmatisation cherchent à résoudre cette difficulté, en replaçant les flexions par leur stem (stemming) ou par leur lemme (lemmatisation).

En recherche documentaire, l'influence du stemming varie selon la langue dans laquelle se fait la recherche, mais, en général, le stemming ne dégrade pas les capacités d'un système de recherche. [Larkey et al., 2005] stipule que dans certaines langues telles que la langue anglaise et française, le stemming favorise le rappel par rapport à la précision, par contre dans les langues qui ont un degré flexionnel élevé telle que l'arabe, le stemming a un effet plutôt positif. Ceci a été démontré dans les travaux de [Aljlayl et Frieder, 2002], qui, en faisant une recherche sur textes radicalisés, ont eu de bons résultats par rapport à une recherche en texte brut.

3.1.4. Représentation basée sur les n-grammes

Une chaîne de caractères consécutifs est appelée n-grammes ou n-uplets [Turenne, 2000]. [Miller et al., 1999] considèrent qu'un n-gramme désignera une chaîne de n caractères consécutifs. Pour un document quelconque, l'ensemble des n-grammes est le résultat obtenu en déplaçant une fenêtre de n cases sur le texte. Le déplacement se fait par étapes d'un caractère et à chaque étape, une photo est prise. L'ensemble de tous les n-grammes est le

résultat de ces photos. Par exemple, pour générer tous les 5-grammes dans la phrase "Je suis un génie", on obtient : je_su, e_sui, suis_, _suis, uis_u, etc.

Il faut noter que certains auteurs comme [Denoyer, 2004] considèrent que l'utilisation des n-grammes est une forme de stemming dans le sens où elle permet de conserver les racines des mots. Le stemming est plus performant mais spécifique à une langue tandis que les n-grammes sont moins performants mais peuvent regrouper des mots de différentes langues issues de la même racine.

3.2. Représentation de documents

La représentation de documents la plus utilisée est sans doute la représentation vectorielle de *G. Salton* [in Jaillet et al., 2003]. Dans ce formalisme vectoriel, chaque dimension de l'espace correspond à un terme d'indexation et la construction du vecteur d'un document est déterminée par des propriétés statistiques de chacun des termes d'indexation du document en question.

3.2.1. Représentation en vecteur binaire

C'est la représentation la plus simple et la plus ancienne pour la représentation des documents. Elle est encore largement utilisée car elle présente un bon compromis entre complexité et performance des systèmes. Dans cette représentation un document est représenté par un vecteur dans l'espace **V** (représentant le vocabulaire) dont les composantes informent sur la présence (valeur égale à 1) ou l'absence (valeur égale à 0) d'un terme dans un document.

D'une manière plus formelle **(1.1)** devient :

$$dj_{binaire} = \begin{cases} wji = 1 & \text{si le ième terme} \in V \text{ apparait dans dj} \\ wji = 0 & \text{sinon} \end{cases} \qquad \textbf{(1.2)}$$

Par exemple, soit à représenter les documents d_1 et d_2 :

d_1 : الجزائر من أكبر الدول العربية مساحة و أوسط تلك الدول من حيث عدد السكان، العربية لغة الجزائر

d_2 : تلعب الجزائر دورا سياسيا هاما بين الدول العربية و الدول الافريقية

V = {دور، سياسي، هام، لعب، لغة، جزائر، أكبر، دول، عربية، افريقية، مساحة، أوسط ، عدد، سكان}.

Ici nous supposons qu'un prétraitement (tokenisation et stemming) est infligé à d_1 et d_2.

Le tableau 1.1 donne la représentation en vecteurs binaires de d_1 et d_2.

	سكان	عدد	أوسط	مساحة	افريقية	عربية	دول	أكبر	جزائر	لغة	لعب	دور	هام	سياسي
d_1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
d_2	0	0	0	0	1	1	1	0	1	0	1	1	1	1

Tableau 1.1 : Exemple de représentation en vecteur binaire

La représentation vectorielle binaire des documents est très simple ce qui est un avantage important pour les systèmes nécessitant un temps de calcul très faible. Son inconvénient majeur est qu'elle est peu informative et ne reflète, parfois pas l'importance d'un terme par rapport à un autre, car elle ne renseigne ni sur la fréquence d'un terme, qui peut constituer une information importante pour, par exemple, un système de recherche d'information, ni sur la longueur des documents.

3.2.2. Représentation en vecteur fréquentiel

La représentation fréquentielle prend en compte le nombre d'apparitions d'un terme dans un document. Ainsi, un document est représenté dans l'espace **V** et chaque composante correspond au nombre d'apparition du terme correspondant dans le document.

Ainsi, w_{ji} dans **(1.1)** devient alors le nombre d'apparitions du terme i dans le document j.

La représentation de d_1 et d_2 de l'exemple précédant deviens alors :

	سكان	عدد	أوسط	مساحة	افريقية	عربية	دول	أكبر	جزائر	لغة	لعب	دور	هام	سياسي
d_1	1	1	1	1	1	2	2	1	2	1	0	0	0	0
d_2	0	0	0	0	1	1	1	0	1	0	1	1	1	1

Tableau 1.2 : Exemple de représentation en vecteur fréquentiel

Normalisation fréquentielle

L'inconvénient majeur de la représentation fréquentielle vient du fait qu'un document de longueur élevée sera représenté par un vecteur dont la norme sera supérieure à celle de la représentation d'un document plus court, ce qui peut avoir une influence négative de certains systèmes sur l'interprétation des représentations des documents tels que le clustering, la recherche documentaire, etc. [Dunoyer, 2004]. Il est donc plus habituel de travailler avec une version normalisée de la représentation fréquentielle où chaque composante du vecteur de représentation code la proportion d'un terme dans le document.

Avec cette normalisation, w_{ji} dans **(1.1)** se calcule alors, en divisant le nombre d'apparition d'un terme i dans un document dj par sa cardinalité.

On obtient alors pour l'exemple de 3.2.1 la représentation dans le tableau 1.3.

	سكان	عدد	أوسط	مساحة	افريقية	عربية	دول	أكبر	جزائر	لغة	لعب	دور	هام	سياسي
d_1	0.10	0,10	0,10	0,10	0,10	0,20	0,20	0,10	0,20	0,10	0	0	0	0
d_2	0	0	0	0	0,125	0,125	0,125	0	0,125	0	0,125	0,125	0,125	0,125

Tableau 1.3 : Exemple de représentation en vecteur fréquentiel normalisé

3.2.3. Représentation en vecteur TF-IDF

Cette représentation est basée sur la loi de Zipf qui décrit la loi de répartition des mots d'un ensemble de documents. Elle tente d'être plus informative que les représentations précédentes en décrivant une relation entre les termes et leurs positions dans un document.

3.2.3.1. Loi de Zipf

La loi de Zipf [Dunoyer, 2004] stipule que les termes les plus informatifs d'un corpus de documents ne sont :

- ni les mots qui apparaissent le plus dans le corpus car ceux-ci sont pour la plupart des mots outils (du type article, mots de liaison, etc.)
- ni les mots les moins fréquents du corpus car ils peuvent, par exemple, être issus de fautes d'orthographe ou de l'utilisation d'un vocabulaire trop spécifique à un unique ou à quelques documents du corpus.

Par contre, un mot qui apparaît beaucoup dans un document possède certainement une information forte sur la sémantique du document.

La figure 1.1 illustre de manière graphique la loi de Zipf. Les deux considérations précédentes ne sont pas antagonistes et peuvent être résumées ainsi, de manière peu formelle : **«***un mot est informatif dans un document s'il y est présent souvent mais qu'il n'est pas présent trop souvent dans les autres documents du corpus***».**

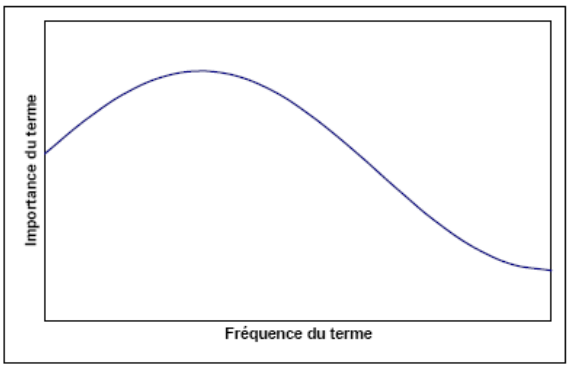


Figure 1.1 : Loi de Zipf

La loi de Zipf illustre l'importance d'un terme en fonction de sa fréquence dans un corpus. Un mot est important s'il n'est ni trop fréquent, ni trop rare.

3.2.3.2. Représentation en vecteur TF-IDF

La représentation en vecteur TF-IDF (*Term Frequency, Inverse Document Frequency*) attribuée à *Gerard Salton*, est certainement la formule de pondération des termes la plus utilisée [Weiss, 2006], elle est aussi bien utilisée pour la recherche documentaire que pour la classification automatique. Elle consiste à multiplier un facteur qui concerne le poids du terme dans le document par un autre qui concerne le poids du terme dans tout le corpus [Salton et Buckley, 1988] :

$$dji_{tf\text{-}idf} = Wji * Wvi \qquad \textbf{(1.3)}$$

dji est la ième composante du vecteur dj

Wji poids du ième terme dans le jème document

Wvi poids du ième terme dans le corpus (v représente le vocabulaire du corpus)

Le modèle le plus classique pour calculer ces deux poids est la fréquence du terme dans le document notée tf_i^d pour le premier et $\log\left(\frac{|D|}{df_i}\right)$ pour le deuxième. |D| et df_i représentent, respectivement, le nombre de documents du corpus et df_i et le nombre de documents qui contiennent le terme i. En résultat nous aurons :

$$dji_{tf\text{-}idf} = tf_i^{dj} * \log\left(\frac{|D|}{df_i}\right) \qquad \textbf{(1.4)}$$

L'exemple de la section 3.2.1 aura la représentation du tableau 1.4.

	سكان	عدد	أوسط	مساحة	افريقية	عربية	دول	أكبر	جزائر	لغة	لعب	دور	هام	سياسي
d_1	0,301	0,301	0,301	0,301	0	0	0	0,301	0	0,301	0	0	0	0
d_2	0	0	0	0	0	0	0	0	0	0	0,301	0,301	0,301	0,301

Tableau 1.4 : Exemple de représentation en vecteur TF-IDF

De cet exemple, nous constatons que les termes سكان، عدد، أوسط، مساحة، أكبر، لغة، لعب، دور، سياسي، هام، sont les plus informatifs.

3.2.3.3. Normalisation de la représentation TF-IDF

Pour ne pas favoriser les documents longs, la représentation en vecteur TF-IDF a été sujette de plusieurs normalisations. L'une de ces normalisations est appelée normalisation en cosinus TFC *(Term Frequency Cosine)* [Jalam, 2003], [Dunoyer, 2004] :

$$dji_{TFC} = \frac{dji_{tf\text{-}idf}}{\sqrt{\sum_{s=1}^{|V|} (djs_{tf\text{-}idf})^2}} \quad \textbf{(1.5)}$$

Il faut noter que dans [Dunoyer, 2004], plusieurs autres pondérations de poids ont été étudiées.

3.2.4. Représentation séquentielle

Cette représentation est une représentation conceptuellement plus simple mais qui nécessite des modèles plus évolués pour pouvoir être utilisée dans différentes problématiques, notamment dans les modèles dynamiques comme les modèles de Markov Cachés. Elle considère qu'un document n'est pas représenté par un vecteur dans un espace donné, mais par une séquence. Ainsi, nous considèrons qu'un document d aura la représentation :

$$d = \left(w_1^d, ..., w_{|d|}^d\right) \quad \textbf{(1.6)}$$

|d| représente le nombre de mots du document d et $w_i^d \in V$ correspond au ième terme.

Cette représentation ne fut utilisée que récemment car sa nature séquentielle nécessite le développement de modèles de recherche d'information plus complexes, en plus si cette représentation était plus informative car elle conserve l'ordre des mots [Dunoyer, 2004], les modèles qui l'utilisaient ne donnaient pas toujours des performances significativement meilleures que des modèles plus simples.

Dans [Jaillet et al., 2003], nous retrouvons une autre méthode de représentation des documents. Au lieu de définir un espace vectoriel où chaque dimension représente un terme d'indexation, souvent assimilé à un stem (radical), l'ensemble des termes est projeté sur un ensemble fini de concepts extrait d'un thesaurus. L'intérêt d'une telle méthode est de réduire les effets polysémiques du vocabulaire. En effet, deux synonymes partageront un ensemble de mêmes concepts. Cette représentation permet donc une factorisation des termes par regroupement de leur champ sémantique.

3.3. Réduction de la dimension du vocabulaire

La dimension de l'espace de représentation ou vocabulaire peut devenir un handicap majeur pour les problèmes de classifications de textes et peut avoir une influence négative sur la précision de la classification. Avec la représentation en sac de mots, chacun des mots d'un corpus est un terme potentiel; or pour un corpus de taille raisonnable, ce nombre peut être de plusieurs dizaines de milliers, d'où la nécessité de la réduction de cette dimension.

D'après [Sebastiani, 2002] il existe deux manières principales de réduire la dimension de l'espace de représentation des textes. La première est dite par **sélection des attributs** *(features selection)* où un score est associé à chaque attribut en fonction d'un algorithme chargé de déterminer son degré de pertinence pour un document donné, les attributs ayant les scores les plus faibles sont éliminés; la deuxième est par **extraction des attributs** *(features extraction)* où un ensemble de nouveaux attributs extérieurs au document sont générés de manière à représenter ce document dans un espace indépendant dont le nombre d'attributs est plus restreint.

3.3.1. Sélection d'attributs

Plusieurs techniques de sélection d'attributs ont été développées en vue de réduire la dimension de l'espace vectoriel ; c'est à dire obtenir un vocabulaire $|V'| << |V|$. Chacune de ces techniques utilise des critères lui permettant de rejeter les attributs jugés inutiles à la tâche de classification en générale. Nous obtenons alors un vocabulaire réduit, des textes représentés par des vecteurs de moindre dimension, un temps de calcul plus abordable et même dans certains cas une précision de classification accrue. Parmi les critères qu'utilisent les techniques de sélection d'attributs, nous retrouvons [Yang et Pederson, 1997] :

La fréquence *(document frequency)*: Il s'agit tout simplement d'éliminer les mots dont le nombre de documents dans lesquels ils apparaissent, est en dessous d'un certain seuil. Ces termes n'auront aucune influence sur la classification.

Le gain d'information *(information gain)*: qui est certainement l'un des plus célèbres critères de sélection d'attributs en apprentissage et surtout ceux basés sur les arbres de décision. On mesure en quelque sorte le pouvoir de discrimination d'un attribut, en sachant sa présence ou son absence.

L'information mutuelle *(mutual information)* : Avec ce critère, on évalue la valeur de l'information mutuelle entre un terme et une classe. Si un terme apparaît souvent dans une classe c'est que l'information mutuelle est élevée, sinon on conclut qu'elle est faible. Une moyenne des scores du terme jumelé à chacune des classes est ensuite calculée.

La faiblesse de cette mesure est qu'elle est beaucoup trop influencée par la fréquence des termes. Pour une même probabilité conditionnelle sachant la classe, un terme rare va être avantagé, car il risque moins d'apparaître en dehors de la classe**.**

La statistique du chi-2 (χ^2) : Est une mesure statistique bien connue, elle évalue le manque d'indépendance entre un terme et un thème (une classe). Elle utilise les mêmes notions de cooccurrence terme/classe que l'information mutuelle, mais à une différence importante, est qu'elle est soumise à une normalisation, qui rend plus comparable les termes entre eux. Cependant, elle perd la pertinence pour les termes peu fréquents.

La force du terme *(term strength)*: La force du terme se propose d'estimer l'importance d'un terme en fonction de sa propension à apparaître dans des documents semblables. Une première étape consiste à former des paires de documents dont la similarité cosinusoïdale est supérieure à un certain seuil. La force d'un terme est ensuite calculée à l'aide de la probabilité conditionnelle, qu'il apparaisse dans le deuxième document d'une paire, sachant qu'il est apparu dans le premier.

3.3.2. Extraction de termes

Toujours dans l'optique de réduire la dimension de l'espace vectoriel et obtenir un vocabulaire $|V'| << |V|$, les techniques d'extractions de termes, à la différence des techniques de sélection, construisent le sous ensemble V' à partir d'une combinaison linéaire des termes de V, pour maximiser la performance de la classification et éliminer les problèmes liés aux synonymies, polysémie et homonymies.

L'une des approches d'extraction de termes est l'**indexation par sémantique latente** *(Latent Semantic Indexing*, LSI*)*, proposée par **[**Deerwester et al., 1990**]** qui utilise une représentation de type conceptuelle. La LSI est fondée sur l'hypothèse d'une structure latente des termes, identifiable par les techniques factorielles, c'est-à-dire, elle tente d'estimer des structures cachées pour générer des termes représentant des concepts. La LSI consiste en une décomposition en valeurs singulières de la matrice dans laquelle chaque document est représenté par la colonne des occurrences des termes qui le composent faisant sortir ainsi les nouveaux termes.

Initialement, cette approche a été utilisée pour effectuer de la recherche d'informations et permet théoriquement de trouver des documents pertinents pour une requête ; même s'ils ne partagent aucun mot avec cette requête. Cette méthode de réduction des dimensions a ensuite été utilisée en entrée des modèles d'apprentissage numérique.

Une autre approche appelée **regroupement de termes** *(term clustering)* testée par *D. Lewis* [in Jalam, 2003], a pour but de grouper les termes qui ont une sémantique commune.

Les groupes (clusters) ainsi créés deviennent les attributs d'un nouvel espace vectoriel. L'idée de départ est que, si deux termes différents apparaissent dans les mêmes classes et dans les mêmes proportions, alors leur réunion en un seul attribut qui affiche la distribution moyenne ne peut pas affecter négativement la performance de la classification.

4. Conclusion

Dans ce chapitre, nous avons essayé de donner une présentation des outils utilisés, généralement, dans le prétraitement de textes *(text preprocessing)* tels que la tokenisation *(tokenization)*, la radicalisation *(stemming)* ainsi que la lemmatisation *(lemmatization)*. Pour chaque outil, nous avons présenté les différentes variantes qui existent actuellement pour les langues européennes telles que l'anglais ainsi que pour la langue arabe. Nous nous sommes attardés sur ceux qui relèvent de la langue arabe vu leur importance dans notre sujet.

Dans la deuxième partie de ce chapitre, nous avons étalé une variété de techniques utilisées actuellement dans le choix de terme de représentation des textes telles que la représentation en sac de mots *(bag of words)*, la représentation par des phrases, la représentation par des stems ou des lemmes, etc.

Pour le codage des termes, des techniques telles que la représentation binaire, la représentation fréquentielle ainsi que la représentation TF-IDF sont exposées.

Enfin, nous avons achevé ce chapitre en donnant un aperçu des techniques destinées à la réduction de dimensions. Nous estimons ainsi avoir présenté une vue globale qui illustre l'importance du prétraitement et la représentation des textes et cela en vu de leur faire subir un clustering documentaire qui sera sujet du chapitre suivant.

Chapitre II
Classification non supervisée documentaire

1. Introduction

D'après une citation de *Brian Everitt* dans [Weiss, 2006];

«*Given a number of objects or individuals, each of which is described by a set of numerical measures, devise a classification scheme for grouping the objects into a number of classes such that objects within classes are similar in some respect and unlike those from other classes. The number of classes and the characteristics of each class are to be determined* **».**

Par analogie, la classification non supervisée documentaire ou clustering documentaire est la tâche qui s'intéresse à trouver, de manière non supervisée et automatique, une organisation cohérente en groupes (clusters) à un ensemble de documents (un corpus) non libellés ou non étiquetés, c'est-à-dire, trouver des regroupements naturels adéquats pour un ensemble de documents sans l'existence préalable de classes libellées. Les composants de chacun de ces groupes possèdent des points de similarité.

2. Problème de la classification documentaire et apprentissage automatique

D'après [Sebastiani, 1999] d'une manière générale, on distingue deux façons d'aborder le problème de la classification automatique. Jusqu'à la fin des années 1980, pour faire une classification documentaire, on construisait un système expert comportant un ensemble de règles définies manuellement, par des experts du domaine, qui ensuite pouvait procéder automatiquement à la classification. Ces règles prenaient généralement la forme d'une implication logique, la présence ou l'absence de certains mots désignait la classe d'appartenance du texte. Cette approche s'avère être très efficace, mais sur un corpus de textes relativement petit, d'où l'inconvénient majeur de cette approche, s'ajoutait à cela l'édition des règles de décision qui s'avérait très longue, la difficulté de l'ajout de nouvelles classes et enfin l'impossibilité de l'utilisation du classificateur dans un domaine autre que

celui sur lequel a été conçu. C'est donc la pertinence de cet ensemble, qui évolue dans le temps, qui a minimisé l'intérêt de cette façon de faire.

Au début des années 1990, une autre approche a pris son envol avec l'avènement de l'apprentissage automatique (*machine learning*), elle s'intéresse plutôt à conférer aux machines la capacité de s'améliorer à l'accomplissement d'une tâche, dans notre cas la classification de textes, en interagissant avec leur environnement. On distingue, deux types d'apprentissages automatiques, apprentissage supervisé et apprentissage non supervisé.

Il est très important de comprendre la différence entre ces deux types. Dans le premier, connu encore sous l'appellation de catégorisation de texte *(Text categorization)* [Sebastiani, 1999], [Sebastiani, 2002], on dispose d'un classificateur déjà entraîné sur une collection de documents labellisés ou étiquetés (modèles) et l'objectif est de classer tout les nouveaux documents non encore labellisés. On se sert d'une collection dite d'entraînement pour entraîner le classificateur pour l'affection des documents à leurs classes. Dans le deuxième cas, communément appelé «clustering documentaire» [Baldi et al., 2003], [Fegas, 2005], [Weiss, 2006], le problème est de regrouper convenablement une collection de documents non étiquetés dans des groupes homogènes (clusters). La labellisation des clusters est ensuite obtenue à partir des documents eux-mêmes d'une façon automatique ou manuelle.

Dans ce qui suit, nous emploierons «clustering» pour désigné la classification non supervisée, nous allons commencé par décrire ce qui est un clustering ainsi que les différents algorithmes utilisés, puis, nous consacrerons la deuxième partie de ce chapitre au clustering documentaire.

3. Clustering

Le terme clustering est utilisé dans la recherche pour décrire les méthodes de regroupement des objets non étiquetés. Plusieurs synonymes peuvent être retrouvés pour le terme clustering tels que apprentissage non supervisé *(unsupervised learning)*, Taxonomy numérique *(numerical taxonomy)*, Quantization vectorielle *(vector quantization)* et Apprentissage par observations *(learning by observation)* [Jain et al., 2000].

En clustering la volonté de **regrouper naturellement** est bien sûr ambiguë [Candillier et al., 2005], elle est le plus souvent formalisée par l'objectif de définir des groupes d'objets tels que la distance entre les objets (selon la mesure de similarité choisie, figure 2.1) d'un même groupe soit minimale, c'est-à-dire, vérifier la propriété de la compacité et que la distance entre groupes soit maximale, c'est-à-dire, les groupes doivent être bornés, autrement

dit, vérifier la propriété de la séparabilité. Notons que ces deux contraintes vont dans deux sens opposés et c'est le meilleur compromis qui doit être trouvé.

D'une manière plus formelle, soit un ensemble X de N objets décrits chacun par leurs P attributs. Le clustering consiste à créer une partition ou une décomposition de cet ensemble en groupes telle que :

Critère 1 : les objets appartenant au même groupe se ressemblent;

Critère 2 : les objets appartenant à deux groupes différents soient peu ressemblants.

Cette vision contraint donc à disposer d'une distance définie sur le langage de description des objets (dans notre cas l'espace de description des documents qui sera un espace vectoriel numérique dans lequel chaque dimension correspond à un attribut distinct).

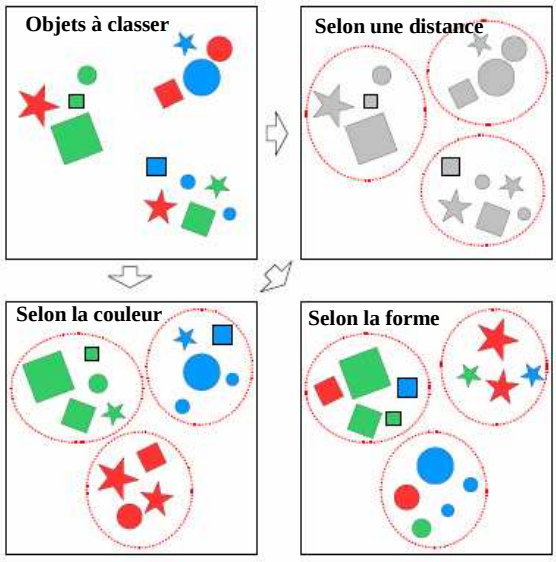


Figure 2.1 : Différents regroupements selon différentes distances

3.1. Différentes étapes dans un processus de clustering

Après avoir représenter les objets à classer, les quatre étapes suivantes définissent un processus typique de clustering.

3.1.1. Définition de la mesure de similarité appropriée au domaine d'application

Cette mesure est définie par une fonction de distance entre deux objets. Plusieurs distances ont été définies dans la littérature [Jain et al., 2000], [Turenne, 2000], en pratique, il peut s'agir de la distance euclidienne ou, au mieux, d'une distance fournie par un expert du domaine. Celui-ci affecte alors un poids à chaque attribut, poids qui traduit l'importance de cet attribut pour le problème considéré.

3.1.2. Regroupement des objets

Ce regroupement peut être effectué par l'une des approches citées dans la section 3.3. Le résultat obtenu peut être un **regroupement en dur** *(Hard clustering)* où chaque objet est affecté à un seul groupe (cluster), ou **regroupement flou** *(fuzzy clustering)* où chaque objet est affecté à plusieurs groupes avec un degré d'appartenance.

3.1.3. Abstraction des données

Consiste à l'extraction d'une simple et compacte représentation pour chaque cluster en vue d'une future analyse, cette abstraction peu être faite par la machine ou par un expert humain. Par exemple, cette représentation peut se résumer en un centroïde [Jain et al., 2000].

3.1.4. Evaluation des résultats

Dans la majorité des cas, la question est de prouver la pertinence ou la signification des clusters obtenus. Quand une approche statistique est utilisée, la validation des résultats ce fait par [Steinbach et al., 1999], [Jain et al., 2000] :

- **un examen interne** *(internal quality)* de la structure de chaque cluster pour évaluer sa densité. Une mesure dite similarité globale *(overall similarity)* est appliquée, elle est égale à la moyenne des similarités entre tout les objets du même cluster pris deux à deux.

- **une évaluation externe** *(external quality)*, qui consiste à comparer la structure obtenue avec une structure élaborée à priori, en utilisant des mesures telles que l'entropie *(Entropy)* ou la F-mesure (*F-measure)*.

Dans la majorité des cas, dans un clustering on ne dispose pas d'une structure élaborée à priori (c'est-à-dire des clusters labellisés) pour évaluer la qualité, d'où le recours généralement à la première technique d'évaluation. Nous reviendrons sur cette technique plus loin dans la section 4.3.

Il faut noter que des auteurs tels que [Jardino, 2005] considèrent que l'évaluation manuelle est une autre possibilité d'évaluer un clustering.

3.2. Mesures de similarité

Dans une classification, les types d'objets traités sont divers (dans notre cas les documents). Ces objets doivent être soigneusement présentés en termes de leurs caractéristiques qui sont principalement le type d'objets (discret, continu ou binaire) et échelle de mesures (Intervalle, nominale, ordinale, …). Ces caractéristiques sont les variables

principales du problème et leur choix influence considérablement les résultats d'un algorithme de classification.

Une fois que les caractéristiques des objets ont été déterminées, on se retrouve confrontés au problème de trouver des moyens appropriés de décider à quelle distance un objet se trouve de l'autre. Les calculs de proximité sont des mesures de la ressemblance entre les paires d'objets. Un calcul de proximité peut mesurer la similarité ou la dissimilarité : plus deux objets se ressemblent, plus leur similarité est grande et plus leur dissimilarité est petite. La ressemblance peut être mesurée par la distance qui existe entre deux objets ou par la relation entre les attributs de ces objets [Weiss, 2006]. Dans le premier cas, et selon le type et la représentation de ces objets, on retrouve plusieurs distances.

Pour les objets qui ont des attributs définis sur une échelle d'intervalle, la distance entre deux objets x et y peut être calculé par les métriques suivantes :

Distance de Minkowski : $d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^q \right)^{\frac{1}{q}}$

Distance euclidienne : $d(x,y) = \sqrt{\left(\sum_{i=1}^{n} |x_i - y_i|^2 \right)}$ (Minkowski, q = 2)

Distance de Manhattan : $d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$ (Minkowski, q = 1)

Distance maximum : $d(x,y) = \max_{i=1}^{n} |x_i - y_i|$ (Minkowski, $q \rightarrow \infty$)

Dans le cas où les attributs sont assignés à des poids, il faut transformer les formules de distance en formules pondérées, par exemple : $d(x,y) = \sqrt{\sum_{i=1}^{k} w_i (x_i - y_i)^2}$, tel que wi est le poids pour l'attribut i.

Les distances citées ci-dessus et bien d'autres sont exposées en détail dans [Berkhin, 2002].

3.3. Techniques du clustering

Depuis la prise en forme des premiers algorithmes de la classification automatique en 1955 [Jain et al., 2000], [Turenne, 2000], [Berkhin, 2002], plusieurs méthodes de clustering telles que les méthodes hiérarchiques, les méthodes de réallocation de type k-moyennes, les méthodes basées sur la densité, les méthodes basées sur les grilles, les méthodes statistiques, les méthodes basées sur la théorie des graphes, les méthodes basées sur la recherche stochastique, les méthodes basées sur les réseaux de neurones, etc. ont été élaborées.

Toutes ces méthodes ainsi que leurs performances diffèrent et dépendent de plusieurs facteurs, nous citons :

- type d'attributs de données traitées
- capacité de traitement d'un gros jeu de données
- capacité de traitement des données de hautes dimensions
- capacité de traiter et réactions aux observations aberrantes
- complexité de l'algorithme de la technique (temps de calcul)
- dépendance de l'ordre des données qui arrivent
- dépendance des paramètres prédéfinis par les utilisateurs
- connaissance a priori sur les données

Dans la littérature, différentes taxonomies ont été données à ces méthodes. Nous nous basons sur celle décrite dans [Jain et al., 2000], dans laquelle deux catégories fondamentales se présentent. La première est celle des méthodes dites hiérarchiques, la deuxième est celle des méthodes dites méthodes de partitionnements. Tandis que les premières établissent graduellement les clusters, les dernières apprennent directement les clusters. C'est-à-dire, qu'elles les découvrent en faisant déplacer des points entre les clusters. Les méthodes de ces deux catégories sont décrites plus loin dans les sections 3.3.1 et 3.3.2, ainsi que leurs variantes.

Pour la taxonomie des méthodes du clustering plusieurs aspects sont pris en compte. Parmi ces aspects nous retrouvons [Jain et al., 2000] :

* **Agglomération contre division**

Cet aspect est en relation directe avec l'aspect opérationnel des algorithmes du clustering. L'approche agglomérative débute avec un objet par cluster et, successivement, les clusters sont fusionnés jusqu'à ce qu'un critère soit satisfait. L'approche divisive débute avec un cluster qui comprend tout les objets, qui sera divisé successivement jusqu'à la satisfaction d'un critère d'arrêt.

* **Monothéique contre polythétique**

Selon que le processus du clustering se fasse avec une prise en charge, séquentielle ou simultané, des attributs des objets, les algorithmes sont soit monothéiques soit polythétiques. C'est-à-dire, pour former les clusters dans les algorithmes polythétiques, touts les attributs des objets sont pris en compte pour calculer la distance, alors que dans les algorithmes

monothéiques, les clusters sont formés en prenant compte le premier attribut puis le deuxième et ainsi de suite.

*** Hard contre Fuzzy**

Dans un clustering en dur *(Hard clustering)* un objet est affecté à un seul cluster, par contre un clustering flou *(fuzzy clustering),* chaque objet est affecté à plusieurs clusters avec un degré d'appartenance, ici une fonction de « *ranking* » est définie pour retourner une valeur comprise entre 0 et 1 pour chaque objet à classer par rapport à chaque cluster.

*** Déterministe contre stochastique**

Cet aspect est très important dans les méthodes par partitionnement, pour l'optimisation de la fonction objective. C'est-à-dire, que l'optimisation se fait en balayant toutes les partitions qui peuvent existées dans le cas déterministe, tandis que dans le cas stochastique quelques partitions sont examinées.

*** Incrémentales contre non incrémentales**

Cet aspect surgit dans le clustering d'un grand ensemble d'objets où la dimensionnalité devient un véritable handicap. Dans le cas incrémentale touts les objets sont pris en compte, tandis que dans le cas non incrémental, une réduction est faite soit sur le nombre d'objets pris en compte soit sur les attributs décrivant les objets.

3.3.1. Méthodes de clustering hiérarchique *(Hierarchical clustering)*

Ces méthodes construisent les clusters graduellement sous une forme hiérarchique, autrement dit, un arbre des clusters qui est appelé un dendrogramme (figure 2.2). Elles sont divisées en 2 types : **Agglomération (ascendante)** et **division (descendante)**.

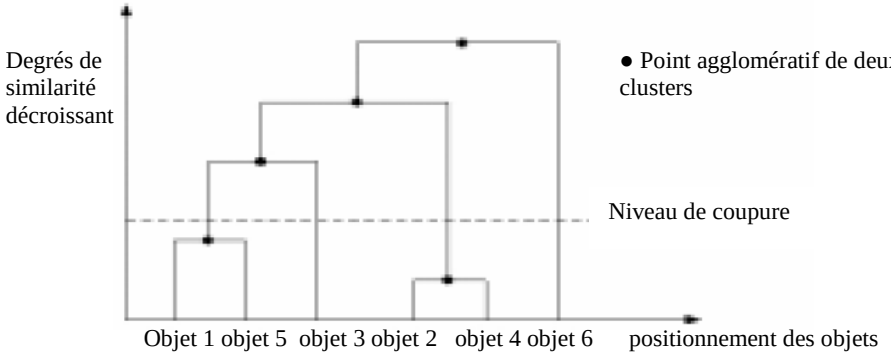


Figure 2.2: Forme type d'un dendrogramme

3.3.1.1. Méthodes hiérarchiques par agglomération (ascendantes)

Les algorithmes de la classification hiérarchique ascendante sont les plus connus des méthodes de la classification automatique [Turenne, 2000], ils sont attribués à *R.R. Sokal* (1963) [in Genane, 2004]. Le processus commence en considérant chaque objet comme étant un cluster et essaye de fusionner deux ou plusieurs clusters appropriés (selon une mesure de similarité) pour former un nouveau cluster. Le processus est itéré jusqu'à ce que tous les points se trouvent dans un même cluster ou bien l'obtention d'un seuil pour lequel on coupe le dendrogramme.

Le fonctionnement de ces méthodes est décrit par l'algorithme 2.1 (au début, les distances entre les objets sont stockés dans une matrice appelée matrice des distances).

1. associer chaque objet à classer à un nouveau cluster;
2. calculer la matrice des distances séparant touts les clusters 2 à 2;
3. tant qu'il existe plus d'un cluster
    - identifier les deux clusters les plus proches et les réunir dans un seul nouveau cluster;
    - mettre à jour la matrice des distances.

Algorithme 2.1 : Algorithme général d'une classification hiérarchique ascendante

Selon la façon avec laquelle les clusters sont fusionnés, plusieurs algorithmes ont été réalisés, nous les retrouvons dans [Turenne, 2000], [Jain et al., 2000], [Bargeton et Devèze, 2005]. Les algorithmes les plus utilisés dans la plupart des méthodes hiérarchiques ascendantes sont les algorithmes du lien simple ou saut minimum (*single link*), les algorithmes du lien ou diamètre complet ou maximal (*complete link*), les algorithmes du lien moyen (*average link*) et les algorithmes de *Ward* ou de la variance minimum (*minimum variance*).

Dans l'algorithme du lien simple, la distance entre 2 clusters est la valeur minimum des distances entre toutes les paires d'objets, l'un du premier cluster, l'autre du deuxième. Une implémentation de cet algorithme est celle de *R. Sibson* (1973) dans SLINK [in Berkhin, 2002], [in Genane, 2004].

Dans l'algorithme du lien complet, la distance entre 2 clusters est la valeur maximale des distances entre toutes les paires d'objets. Une implémentation de cet algorithme est celle de *D. Defays* (1977) dans CLINK [in Berkhin, 2002].

Dans l'algorithme du lien moyen, la distance entre 2 clusters est la valeur moyenne des distances entre toutes les paires d'objets, l'un du premier cluster, l'autre du deuxième.

Une implémentation de cet algorithme est celle de *E.M. Voorhees* (1986) dans *Voorhees's method* [in Berkhin, 2002].

Dans l'algorithme de la variance minimum, la distance entre 2 clusters est calculée selon la distance de *Ward* (équation 2.1). Une implémentation de cet algorithme est celle de *J. H. Ward* (1963) [in Turenne, 2000], [in Berkhin, 2002].

$$\delta(a,b) = \frac{P_a P_b}{P_a + P_b} d^2(g_a, g_b) \qquad \textbf{(2.1)}$$

P et g représentent, respectivement, le poids et le centre de gravité des deux clusters a et b.

L'algorithme du lien complet produit des clusters étroitement liés ou compacts [Jain et al., 2000], en revanche, l'algorithme du lien simple souffre d'un effet d'enchaînement, il a une tendance à produire des clusters qui sont prolongés (figure 2.3). Cependant l'algorithme du lien simple est plus souple que l'algorithme du lien complet. D'un point de vue pratique, on a observé que l'algorithme du lien complet produit des hiérarchies plus utiles dans beaucoup d'applications que l'algorithme du lien simple.

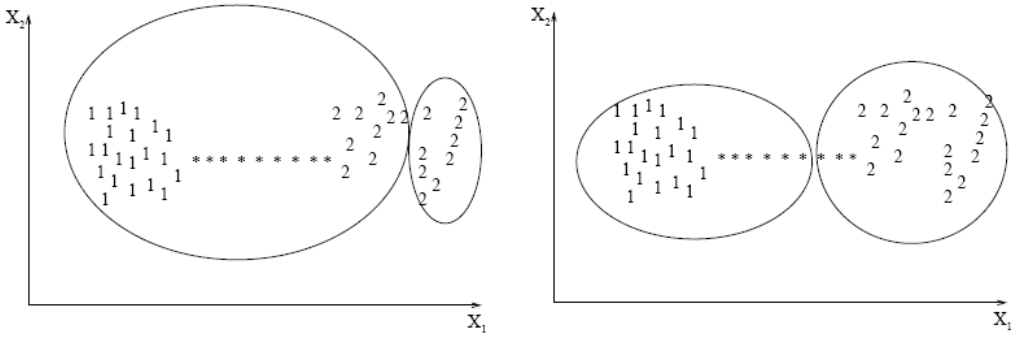


Figure 2.3 : Clustering avec lien simple (gauche) et lien complet (droite) d'objets contenant 2 classes 1 et 2 avec le bruit *

Pour les algorithmes du lien moyen, il y a une rigueur intermédiaire entre les algorithmes du lien complet et ceux du lien simple [Turenne, 2000]. En ce qui concerne les algorithmes de *Ward*, on a constaté qu'ils conduisent, généralement, à des structures de clusters exactes, en revanche, ils sont sensibles aux bruits et pauvres pour retrouver les clusters allongés.

3.3.1.2. Méthodes hiérarchiques par division (descendantes)

En considérant tous les points comme un seul cluster au début, le processus divise successivement les clusters en clusters plus raffinés [Chavent et al., 1999], [Turenne, 2000].

Le processus continu jusqu'à ce que chaque cluster contienne un seul point ou bien l'atteinte d'un nombre désiré de clusters.

En d'autre terme, soit I l'ensemble des objets, I est divisé en 2 clusters C_0 et C_1 ; puis ces deux clusters sont, à leurs tours et selon une distance du maximum, scindés en deux autres clusters, C_0 en C_{00} et C_{01}, C_1 en C_{10} et C_{11} et ainsi de suite jusqu'à l'obtention des clusters à un objet. Le fonctionnement de ces méthodes est décrit dans l'algorithme 2.2.

1. calcul des distances sur I et tri des valeurs par ordre décroissant;
2. évaluation de la distance max $d(i_1, i_0)$ où chaque élément de I est associé à un cluster C_0 représenté par i_0 ou C_1 représenté par i_1 ;
3. on considère le cluster Ci qui possède le diamètre maximum (distance maximum). On divise C_i en C^a_i et C^b_i en attribuant chaque élément de C_i soit à C^a_i soit à C^b_i;
4. recalcul des diamètres par ordre décroissant et s'arrêter dès que le nombre de clusters = Card[I].

Algorithme 2.2 : Algorithme général d'une classification hiérarchique descendante

Ils existent deux principales catégories de ce type de méthodes, la première comporte les méthodes polythétiques telle que la méthode de *MacNaughton-Smith et Coll* (1964), la deuxième comporte les méthodes monolithiques telle que la méthode de *Williams et lambert* (1959) [in Chavent et al., 1999].

3.3.1.3. Avantages et inconvénients

En général, les méthodes hiérarchiques ont l'avantage d'avoir une flexibilité pour le niveau de granularité (on peut atteindre un cluster le plus fin ou le plus épais comme souhaité), la capacité de traiter n'importe quelle mesure de similarité ou distance et l'application sur n'importe quel type d'attributs. En contre partie, elles sont très coûteuses en temps et en espace mémoire, en plus elles ont un critère de terminaison qui est souvent vague.

3.3.2. Méthodes de clustering avec partitionnements ou à plat (*Partitional clustering*)

Ces méthodes produisent une seule partition d'objets au lieu d'une structure de clusters, en d'autres termes, elles cherchent à diviser la population initiale en groupes disjoints [Ah-Pine et al., 2005], [Weiss, 2006], en optimisant une fonction objective qui est définie d'une façon locale (sur un sous-ensemble d'objets) ou globale (sur tous les objets) [Jain et al., 2000].

Les méthodes de type k-moyennes *(k-means)* sont les plus connues en clustering avec partitionnements [Jain et al., 2000]. L'algorithme original pour ces méthodes est celui développé par *J. McQueen* en 1967 (figure 2.4) [in Faber, 1994], [in Fegas, 2005], bien que

de nombreux travaux de *Thorndike* et *Forgy* aient été menés parallèlement et indépendamment pour introduire des variantes ou des généralisations.

L'algorithme de *McQueen* est décrit par l'algorithme 2.3.

1. choisir aléatoirement k objets (centres) qui représentent les K clusters initiaux;
2. affecter les objets aux clusters. Pour chaque objet x, le centre qui lui est assigné est celui qui lui est le plus proche, selon une mesure de distance (habituellement la distance euclidienne est utilisée);
3. une fois que tous les objets sont affectés, recalculer les centres des k clusters;
4. répéter les étapes 2 et 3 jusqu'à ce que plus aucune réaffectation ne soit possible.

Algorithme 2.3 : Algorithme général d'une méthode de type k-moyennes

En général, on retrouve trois principales catégories de ce type d'algorithmes [Pasquier, 2003] :

k-moyennes : un cluster est représenté par un centre de gravité.

k-médoïds : un cluster est représenté par un objet.

Nuées dynamiques : un cluster est représenté par un noyau composé d'objets centraux.

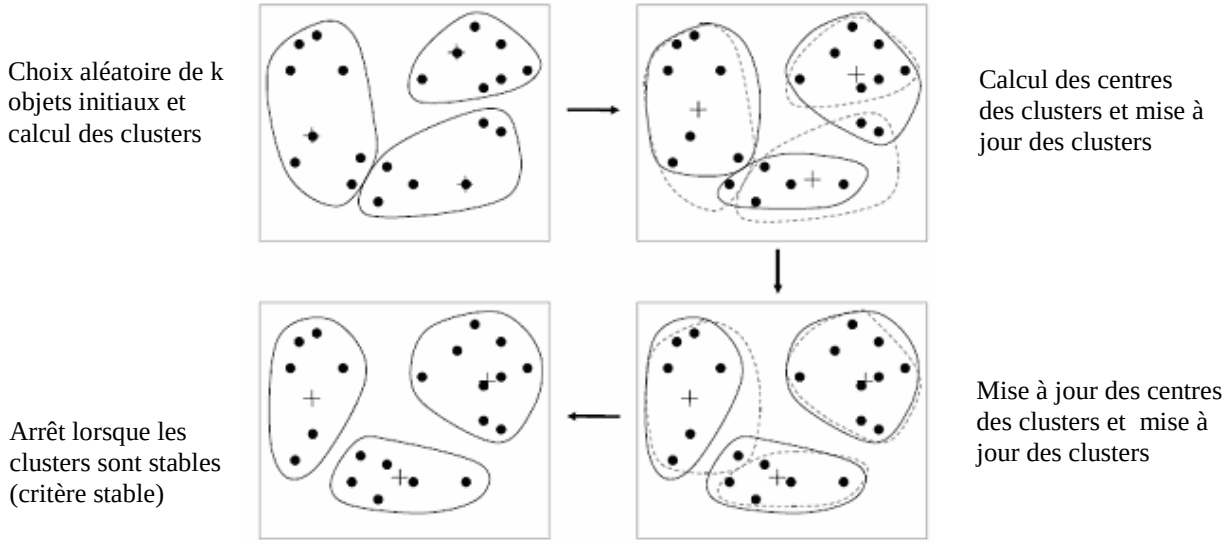


Figure 2.4 : Différentes étapes d'une méthode de type des k-moyennes (k-means)

3.3.2.1. k-moyennes ou centres mobiles *(k-means)*

Ce type d'algorithmes est le plus connu dans la communauté de classification des données. Dans ce type, chaque cluster est représenté par une moyenne *(mean)* ou une moyenne pondérée nommée «centroïde» [Steinbach et al., 1999] qui est calculée selon l'équation suivante :

$$C_S = \frac{1}{|S|} \sum_{X_i \in S} X_i \qquad \textbf{(2.2)}$$

C_S est le centroïde du cluster S et X_i un élément du cluster.

Le fonctionnement de ce type d'algorithme est celui décrit dans l'algorithme 2.3, il commence avec un ensemble de K objets de référence (germes) choisis par l'utilisateur. Au début, les objets restants sont partitionnés sur les K clusters formés autour des germes, les centroïdes sont ensuite calculés. Un objet appartient à un cluster, si le centroïde de ce cluster est le plus proche de lui. La mise à jour des centroïdes et l'affectation des objets aux clusters sont réalisées pendant les itérations successives.

Plusieurs variantes des méthodes de type k-moyennes ont été élaborées, deux critères font la différence entre ces versions :

- comment se fait la mise à jour des clusters
- quand est ce que se fait la mise à jour

Pour le premier critère, les algorithmes de ce type diffèrent dans le détail de la génération et de l'ajustement des clusters. Selon [Faber, 1994], il y a 3 algorithmes de base pour ce type : Standard k-means, l'algorithme de Lloyd et Continuous k-means.

Algorithme de Lloyd : Attribué à *S. P. Lloyd* [in Kanungo et al., 2002], l'initialisation de l'algorithme est similaire à la description ci-dessus. Les ajustements sont réalisés en calculant le centroïde pour chaque cluster et en utilisant ces centroïdes comme les points de référence dans l'itération suivante pour tous les objets à classer. La mise à jour des centroïdes n'est faite qu'après une itération.

Standard k-means : Attribué à *J. McQueen* [in Faber, 1994], cet algorithme est meilleur que celui de Lloyd en terme de l'utilisation plus efficace de l'information à chaque pas d'itération, c'est-à-dire que la mise à jour des clusters et des centroïdes est faite pendant et après une itération.

Continuous k-means : Attribué à *J. McQueen* [in Faber, 1994], dans cet algorithme, à la différence des algorithmes de Lloyd et standard k-means où les points de référence initiaux sont arbitrairement choisis, ces points sont choisis à partir d'un échantillon aléatoire de la population entière. En plus et contrairement au standard k-means, où touts les objets sont séquentiellement examinés, cet algorithme n'examine qu'un échantillon aléatoire des objets.

Si le jeu de données est gros et l'échantillon est représentatif du jeu de données, alors l'algorithme peut converger plus vite qu'un algorithme qui doit examiner séquentiellement tous les objets.

Pour le deuxième critère, deux variantes de l'optimisation itérative des k-means existent [Berkhin, 2002] :

Algorithme de Forgy : Attribué à *E. Forgy*, les itérations, dans ce type d'algorithme, disposent de deux pas, réaffectation de chaque objet au centroïde le plus proche puis recalcule des centroïdes des nouveaux clusters créés. Les itérations continuent jusqu'à ce qu'on atteigne un critère de terminaison (par exemple, il n'y a plus de réaffections).
Le principal avantage de cet algorithme est l'insensibilité à l'ordre des données.

Algorithme d'optimisation itérative : réaffecte les points en se basant sur une analyse plus détaillée des effets sur la fonction objective quand un point est déplacé de son cluster à un cluster potentiel. Si l'effet est positif, ce point sera réaffecté et deux centroïdes seront recalculés.

3.3.2.2. K-médoïds

Dans ce type d'algorithmes, un cluster est représenté par un de ses objets appelé médoïd qui minimise la somme des distances aux autres objets du même cluster selon :

$$\min \sum_{X_i \in S} d(M_S, X_i) \qquad \textbf{(2.3)}$$

M_S est le médoïd du cluster S et X_i un élément du cluster.

Une telle représentation nous donne deux avantages. Elle s'adapte à n'importe quel type d'attributs et le médoïd est choisi comme une fraction des objets prédominants dans un cluster, donc il n'est pas sensible aux aberrants. Si les médoïds sont choisis, les clusters sont définis comme les sous-ensembles des objets proches du médoïd correspondant, et la fonction objective sera définie comme la distance moyenne (ou d'autres mesures de similarité) entre un objet et le médoïd. L'algorithme phare de ce type est celui développé par *L. Kaufman* et *P. Rousseeuw* (1990) [in Ng et Han, 1994] intitulé PAM, ensuite deux autres améliorations de cet algorithme ont apparu : CLARA et CLARANS.

PAM *(Partitioning Around Medoids)* développé par *L. Kaufman* et *P. Rousseeuw* (1990) [in Ng et Han, 1994], son fonctionnement est décrit par l'algorithme 2.4.

1. Choisir aléatoirement k objets (médoïds) qui forment les K clusters initiaux;
2. Affecter les N-k objets aux k clusters. Pour chaque objet x, le médoïd qui lui est assigné est celui qui lui est le plus proche, selon une mesure de distance;
3. Une fois tous les objets affectés
    _Choisir aléatoirement un non-médoïd M_R;
    _Pour chaque médoïd M_j
        Calculer le coût du remplacement CR de M_j par $M_{R;}$
        Si CR < 0 Alors
            Remplacer M_j par M_R;
            Réaffecter tous les objets qui n'ont pas été sélectionnés aux k clusters;
4. Répéter l'étape 3 jusqu'à stabilisation des clusters;

Algorithme 2.4 : Algorithme général de PAM

Le coût des remplacements est calculé selon la formule de l'erreur carrée suivante :

$$CR=E(M_R)-E(M_j) \qquad \textbf{(2.4)}$$

$$E=\sum_{i=1}^{k}\sum_{X\in C_i} d(X,M_i)^2$$

CLARA ***(Clustering LARge Applications)*** développé aussi par *L. Kaufman* et *P. Rousseeuw* (1990) [in Ng et Han, 1994], l'amélioration de CLARA par rapport à PAM est qu'il ne se base pas sur l'ensemble entier d'objets, il travaille sur des échantillons d'objets. Une petite partie d'objets est prise aléatoirement pour représenter tout l'ensemble, puis, PAM est déroulé sur cet échantillon pour déterminer les K médoïds. L'algorithme est exécuté sur plusieurs échantillons pour obtenir le meilleur résultat. En conséquence, CLARA peut traiter un plus gros jeu de données que PAM.

Un inconvénient de cet algorithme est que si un point qui serait le meilleur médoïd n'apparaît dans aucun échantillon, alors CLARA ne trouvera jamais le meilleur résultat.

CLARANS ***(Clustering Large Applications based on RANdomized Search)*** développé par *R.T. Ng* et *J. Han* [Ng et Han, 1994], CLARANS est une combinaison de PAM et CLARA. Dans cet algorithme, on a utilisé une abstraction de graphe pour représenter le problème de recherche des k meilleurs médoïds. On construit un graphe dont chaque noeud est un ensemble de k objets. Deux noeuds sont dits voisins s'ils ne diffèrent que par un seul point. En conséquence, le problème de déterminer un ensemble de k meilleurs médoïds devient le problème de recherche dans ce graphe du meilleur noeud. Le critère pour estimer qu'un noeud est meilleur qu'un autre est celui utilisé dans PAM. Pour déterminer le meilleur nœud, on utilise l'algorithme CLARA pour minimiser l'espace de recherche.

En conséquence, cela ne donne pas toujours un bon résultat car il travaille sur une région locale.

3.3.2.3. Nuées dynamiques

Cette méthode est attribuée à *E. Diday* [Diday, 1971], le centre d'un cluster est représenté par ensemble d'objets réels appelé noyau supposé plus représentatif que le centre de gravité.

Pour déterminer les éléments d'un noyau, une fonction dite d'agrégation-écartement (équation 2.5) [Diday, 1971], [Turenne, 2000] est utilisée pour qualifier l'appartenance d'un objet i au noyau m d'un cluster C pour l'ensemble des noyaux M :

$$R_1(i,C,M) = \frac{D(i,m) * D(i,C)}{\left(\sum_{n \in M} D(i,n)\right)^2} \quad \text{ou} \quad R_2(i,C,M) = D(i,C) \qquad \textbf{(2.5)}$$

$D(i,P) = \sum_{j \in P} d(i,j)$ représente la distance d'un objet i à une partition P recherchée;

Les éléments de m sont appelés les étalons du cluster C;

D(i,m) aura pour effet d'**agréger** les étalons;

D(i,C) aura pour effet de ramener les étalons vers le centre de leur cluster;

$\sum_{n \in M} D(i,n)$ aura pour effet d'**écarter** les éléments de n entre eux.

En suite on, retient les objets qui ont un comportement stable pendant plusieurs itérations et qui seront appelés formes fortes.

3.3.2.4. Avantages et inconvénients

L'avantage principal des algorithmes de type k-moyennes est leur simplicité [Fegas, 2005] qui est à l'origine de leur popularité, en plus ce type fonctionne bien avec les grandes masses d'objets à classer à l'inverse de l'approche hiérarchique dans laquelle la construction des clusters avec une hiérarchie (dendrogramme) est très coûteuse. Des heuristiques sont utilisées pour l'optimisation d'itérations sous la forme de mécanisme de réallocation qui réaffectent les objets entre les clusters. Il raffine graduellement les clusters et donc peut donner des clusters de meilleure qualité.

En contre partie, les algorithmes de type k-moyennes présentent pas mal d'inconvénients. En premier lieu, le résultat obtenu par ce type est fortement dépendant de la partition initiale qui, quant à elle, n'est pas facile à déterminer; donc il faut les exécuter plusieurs fois avec différents états initiaux afin d'obtenir un meilleur résultat. En plus le

processus de clustering est sensible aux aberrants, s'ajoute à cela aussi que ce type d'algorithmes n'est pas extensible et il ne travaille que sur des données numériques.

Pour remédier à ces inconvénients, certaines améliorations et extensions ont été proposées [in Jain et al., 2000], [in Berkhin, 2002]. Parmi ces améliorations notant l'algorithme **ISODATA** de *G. H. Ball* et *D. J. Hall* (1965), qui permet de déterminer automatiquement un nombre initial de clusters et dont le résultat est acceptable; les algorithmes des **K-modes** et **K-prototypes** pour manipuler les données catégories développés par *Z. Huang* (1998); l'algorithme pour l'accélération des k-moyennes par l'**inéquation triangulaire;** l'algorithme **single pass k-means** pour travailler sur un gros jeu de données développé par *P. Bradley, U. Fayyad et C. Reina* (1998), etc.

4. Classification non supervisée documentaire *(Document Clustering)*

Comme il a été mentionné auparavant, le clustering est le processus qui consiste à retrouver des groupes naturels d'objets à classer, selon une mesure de similarité. En clustering documentaire, les objets à classer sont des documents représentés sous une même forme (par exemple sac de mots) en utilisant une des techniques de clustering.

Le besoin d'une telle classification, supervisée ou non, est expliqué par le très grand nombre de textes qu'une base documentaire peut contenir et donc d'une part, la difficulté d'effectuer une recherche documentaire et d'autre part, l'organisation et l'exploration de la structure de cette base et cela en temps et en efficacité [Weiss, 2006]. Dans ce qui suit nous nous intéresserons à la recherche documentaire locale qui est l'un des domaines qui peut être, à notre avis, améliorée par une classification non supervisée et surtout si la base documentaire dans laquelle se fait la recherche est en langue arabe.

4.1. Recherche documentaire *(Document retrieval)*

La recherche documentaire consiste à trouvés les documents '**pertinents**'[1] (*relevant documents*) recherchés en réponse à une requête introduite par un utilisateur, d'ailleurs, c'est la vocation principale de ce que nous appelons moteurs de recherche (*search engines*).

A la fin des années 60, *G. Salton* (1968) [in Bellot et El-bèze, 2000] préconisait l'emploi de méthodes globales pour regrouper les documents des bases documentaire et permettre une recherche plus rapide en ne calculant plus les distances entre la requête et chaque document mais seulement entre la requête et chaque cluster, *«Clearly in practice it is*

[1] Certains auteurs préfèrent employer le terme ressemblance entre documents et requête.

not possible to match each analysed document with each analysed search request because the time consumed by such operation would be excessive». Le besoin de cette approche est expliqué par la limitation en puissance de calcul des ordinateurs de cette époque pour le calcul de la distance entre chaque document et la requête en question. Il faut noter qu'ici nous nous intéressons beaucoup plus à la recherche sur des bases documentaire locales.

Avec le développement technologique du matériel informatique et l'augmentation du volume des bases documentaires, les chercheurs se sont retournés vers une approche qui s'appuie sur le calcul du degré de similarité (ressemblance), mesuré par la distance entre chaque document de la collection et la requête. Cette distance est utilisée pour donner un rang *(rank)* à chaque document qui va servir pour son classement dans la liste retournée à l'utilisateur.

4.1.1. Modèles de systèmes de la recherche documentaire

La présentation des documents pertinents à l'utilisateur se fait, généralement, sous forme d'une liste. Malheureusement, cette liste ne comporte pas toujours les documents pertinents, ces derniers se trouvent souvent 'noyés' dans un ensemble de documents non pertinents. Le problème revient donc, à dire quel est le mécanisme adéquat pour déterminer la pertinence d'un document par rapport à une requête ? Pour cela, plusieurs modèles de systèmes de recherche documentaire ont été conçus. Ces modèles sont répertoriés sur trois catégories fondamentales [Zhai, 2002].

4.1.1.1. Modèle basé sur la similarité *(Similarity-based Model)*

Dans ce type de modèle, la pertinence d'un document est fortement corrélée avec la similarité entre ce document et la requête. Plus le document est similaire à la requête plus il est pertinent. Le modèle vectoriel est le plus connu dans ce type.

4.1.1.2. Modèle de pertinence probabiliste *(Probabilistic Relevance Model)*

Dans ce type de modèle, on s'intéresse plutôt à la question «Quelle est la probabilité pour que ce document soit pertinent pour cette requête?». Ici, le système de recherche n'est pas en mesure de déterminer le degré de la pertinence d'un document, pour cela, il utilise une variable aléatoire binaire pour estimer le degré de pertinence d'un document. Pour la détermination de la valeur de cette variable, plusieurs modèles, tels que le modèle discriminant, le modèle polynomial, etc. ; ont été étudiés [Zhai, 2002].

4.1.1.3. Modèle d'inférence probabiliste *(Probabilistic inference Model)*

Dans ce type de modèle, on tente d'inférer ou de prouver la requête à partir du document ou vis versa, pour estimer l'incertitude de la pertinence du document. Pour inférer ou prouver une requête à partir d'un document plusieurs modèles ont étalés dans [Zhai, 2002].

4.1.2. Clustering documentaire appliqué à la recherche documentaire

Deux principaux axes se sont distingués pour l'application d'un clustering documentaire pour l'amélioration de la précision d'une recherche documentaire.

4.1.2.1. Application du clustering à tous les documents de la collection

Initiée par *G. Salton* (1968) [in Bellot, 2000] et reformulée par [Rijsbergen, 1979]. En se basant sur le fait que «les documents similaires sont pertinents pour les mêmes requêtes[1]», un clustering est appliqué sur la collection documentaire, pour regrouper les documents homogènes, avant le lancement d'une recherche sur celle-ci. Quand l'utilisateur introduit sa requête, l'algorithme de recherche renvoie les documents qui sont assortis à la requête introduite. Il faut noter que l'organisation (en cluster) de la collection n'est pas exposée à l'utilisateur.

4.1.2.2. Application du clustering à la liste retournée par le système de recherche

Plusieurs travaux se sont illustrés sur cet axe, nous mentionnons ceux de [Bellot et El-bèze, 2000], qui ont effectué un clustering documentaire sur la liste retournée par un système de recherche suivi d'une reconstitution d'une nouvelle liste basée sur les clusters retournés.

4.1.3. Processus de recherche documentaire

Généralement, un processus de recherche documentaire en texte intégral se décompose en trois étapes principales [Bellot, 2000] :

4.1.3.1. Préparation et indexation des textes sur lesquels porte la recherche

Durant cette phase, un prétraitement, tel que décrit dans le chapitre I, est appliqué aux textes de la base documentaire. Cela donne lieu à la création d'un index. Le même prétraitement est appliqué à la requête introduite en langage naturel par l'utilisateur.

4.1.3.2. Recherche proprement dite

Ici il s'agit de mesurer, grâce aux informations enregistrées dans l'index, la similarité entre chaque texte de la collection et la requête au moyen d'une distance telle que la mesure du cosinus (*cosine)* décrite dans la section 4.2, chapitre 2.

[1] Observé par Keith Van Rijsbergen «*closely associated documents tend to be relevant to the same requests*»

4.1.3.3. Présentation des résultats de la recherche

Pour terminer, les documents trouvés sont ordonnés en fonction de leurs rangs sous forme d'une liste qui est proposée à l'utilisateur.

4.1.4. Evaluation des systèmes de recherches documentaires

L'évaluation des systèmes de recherches documentaires est un problème complexe, d'autant plus qu'elle doit souvent prendre en compte l'utilisateur [Gallinari et al., 1999]. Les deux mesures les plus employées dans ce domaine sont la Précision *(Precision)* et le Rappel ou Couverture (*Recall*) [Rijsbergen, 1979]**.**

4.1.4.1. Précision *(Precision)*

Certains documents retenus par le système peuvent ne pas correspondre à la demande de l'utilisateur, la précision mesure la capacité d'un système de recherche à ne pas juger comme pertinent un document qui ne l'est pas. Elle calculée selon l'équation (2.6).

$$\text{Précision} = \frac{DP}{DR} \qquad \textbf{(2.6)}$$

DP est le nombre de documents pertinents ramenés par le système de recherche

DR est le nombre de documents ramenés.

On utilise aussi la notion de bruit (Figure 2.5) qui présente le problème selon le point de vue opposé. Le bruit est le pourcentage de textes non pertinents renvoyés par le système **[**De Loupy, 2000**]** :

$$\text{Bruit} = 1 - \text{précision} \qquad \textbf{(2.7)}$$

4.1.4.2. Rappel ou couverture (*Recall*)

Le système de recherche peut considérer certains textes comme non pertinents alors qu'ils correspondent à la requête de l'utilisateur. Le rappel (équation 2.8) mesure la capacité d'un système de recherche à détecter les documents pertinents.

$$\text{Rappel} = \frac{DP}{DPT} \qquad \textbf{(2.8)}$$

DP est le nombre de documents pertinents ramenés par le système de recherche

DPT le nombre total de documents pertinents contenu dans le corpus

On utilise aussi la notion de silence (Figure 2.5) qui est le point de vue opposé. Le silence est le pourcentage de textes pertinents non renvoyés par le système **[**De Loupy, 2000**]** :

$$\text{Silence} = 1 - \text{rappel} \qquad \textbf{(2.9)}$$

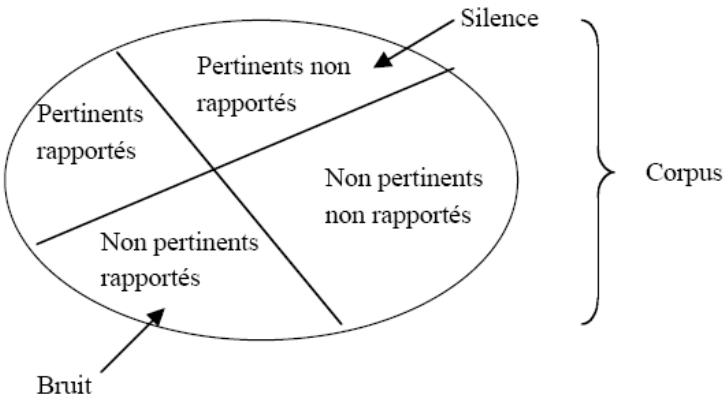


Figure 2.5 : Représentation du bruit et du silence

4.1.4.3. Courbe Précision-Rappel

Le comportement d'un système est en général donné par une courbe dite de **Précision-Rappel** qui donne les valeurs de la précision et du rappel pour différents seuils. On distingue deux types de courbes :

*** Courbe interpolée**

Dans la courbe interpolée, la précision est calculée pour des valeurs prédéfinies du rappel de 0 à 1 par pas de 10 % (figure 2.6). En pratique, ces valeurs peuvent ne pas être atteintes exactement, les valeurs de la précision correspondantes doivent être alors interpolées.

La règle d'interpolation générale, utilisée au cours des compagnes TREC [Bellot, 2000], est la suivante :

- La précision pour un niveau de rappel i est la précision maximale obtenue pour un rappel supérieur ou égal à i.

- Pour un rappel nul, la précision est celle qui correspond au niveau maximal de précision obtenu pour un rappel quelconque.

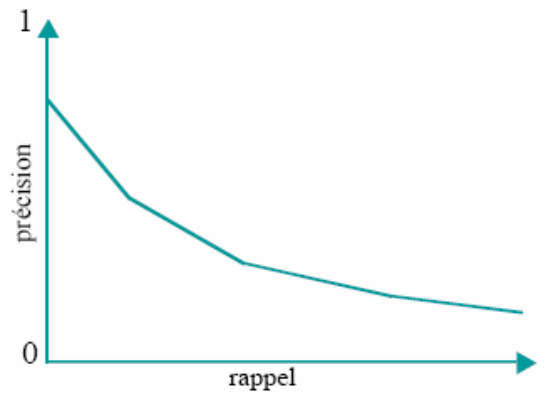


Figure 2.6 : Courbe d'interpolation Rappel/Précision

***Courbe non interpolée**

Il est possible de calculer la précision et le rappel correspondant à chaque valeur du seuil des documents rapportés et de tracer l'évolution de ces deux mesures indépendamment (figure 2.7). Par exemple, la précision et le rappel des 5 premiers documents **«quelle est la proportion de documents pertinents parmi les cinq premiers?»**, puis les 10 premiers documents, etc. Dans la campagne TREC [Bellot, 2000], les différents niveaux testés ont pour valeurs : 5, 10, 15, 20, 50, 200, 500 et 1000.

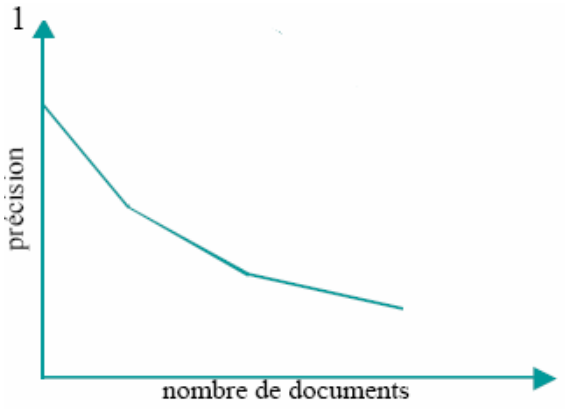


Figure 2.7 : Courbe de la précision des n premiers documents

D'autres mesures de performance sont également utilisées telles que la F-mesure *(F-measure)* initiée par [Rijsbergen, 1979], qui une combinaison entre la précision et le rappel. Elle est calculée selon l'équation 2.10.

$$F-\text{mesure} = \frac{2 \bullet \text{Précision} \bullet \text{Rappel}}{\text{Précision} + \text{Rappel}} \qquad \textbf{(2.10)}$$

4.1.5. Travaux connexes en recherche documentaire en langue arabe

Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970. Les premiers travaux concernaient notamment les lexiques et la morphologie ainsi que la recherche documentaire. Les travaux qui relevaient de la recherche documentaire, visant à améliorer la précision, se sont focalisés surtout sur le stemming. Dans ce qui suit nous essayerons d'exposer les principaux travaux relatif à cet axe qui, selon nos investigations, ont débuté avec l'expérimentation de [Al-Kharashi et Evens, 1994] qui ont appliqué 29 requêtes sur un corpus 355 textes. L'expérimentation s'est déroulée en utilisant des mots, des stems et des lemmes dans les indexes. Ils ont constaté que l'utilisation des stems et des lemmes améliore considérablement le rappel. D'autres travaux tels que [Aljlayl et Frieder, 2002], [Xu et al., 2002], [Larkey et al., 2002], [Larkey et al., 2005] se sont aussi intéressés à développer des stemmers pour l'amélioration de la recherche documentaire.

Aussi, nous recensons aussi les travaux de [Darwish et Oard, 2002] qui ont effectué une recherche documentaire sur un corpus en langue arabe en introduisant des requêtes en anglais. *S. Boulaknadel* (2005) a utilisé des syntagmes nominaux (phrase statistiques) pour améliorer les performances et la précision d'un système de recherche d'informations sur une collection de documents arabes spécialisés dans le domaine de l'environnement [Boulaknadel, 2005].

4.2. Similarité entre les documents *(documents similarity)*

La similarité entre deux documents, représentés par leurs vecteurs respectifs dans un espace vectoriel, est calculée à l'aide d'une corrélation quelconque entre les deux vecteurs. Une telle corrélation peut être l'une des deux mesures largement utilisées et qui sont le cosinus de l'angle formé par les deux vecteurs ou encore la distance euclidienne (section 3.2) [Dubois, 2002], [Weiss, 2006].

La mesure du cosinus est une technique qui découle du faite que, si deux vecteurs ont approximativement, les mêmes attributs alors ils pointent vers la même direction dans l'espace de représentation (figure 2.8). Donc pour calculer la similarité entre deux documents représentés par leurs deux vecteurs d_i et d_j en utilisant le cosinus [*G. Salton*, 1983] [in Bellot, 2000], [in Preux, 2006], [in Weiss, 2006], on a :

$$sim(d_i, d_j) = \cos(\alpha) = \frac{d_i . d_j}{|d_i||d_j|} = \frac{\sum_k d_{ik} . d_{jk}}{\sqrt{\sum_k d_{ik}^2 . \sum_k d_{jk}^2}} \qquad \textbf{(2.11)}$$

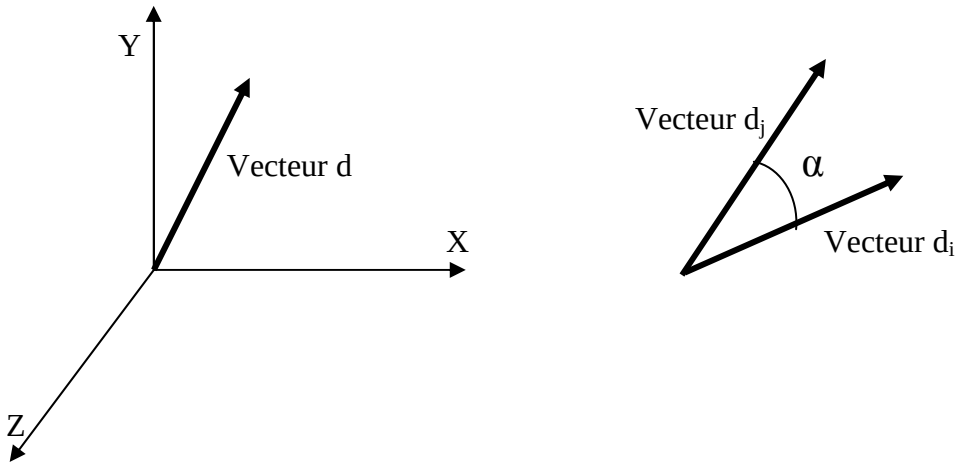


Figure 2.8 : Représentation et Similarité entre deux documentes dans l'espace

4.3. Evaluation du clustering documentaire

Soit l'ensemble des documents contenus dans cluster S ainsi que leurs vecteurs de représentation. On définie le centroïde du cluster S selon l'équation 2.12.

$$C_S = \frac{1}{|S|}\sum_{d \in S} d \qquad \textbf{(2.12)}$$

Comme il a été mentionné dans la section 3.1.4, une des techniques de mesures utilisées pour évaluer la qualité des clusters générés est l'examen interne (*internal quality)* dans lequel on utilise une mesure globale *(overall similarity)* qui est égale à la similarité entre touts les documents du même cluster pris deux à deux [Steinbach et al., 1999]. Pour cela, on utilise généralement, une similarité moyenne pondérée.

$$Q_S = \frac{1}{|S|^2}\sum_{\substack{d_i \in S \\ d_j \in S}} sim(d_i, d_j) = \frac{1}{|S|}\sum_{d_i \in S} d_i \bullet \frac{1}{|S|}\sum_{d_j \in S} d_j \qquad \textbf{(2.13)}$$

d_i et d_j sont deux documents du cluster S

$sim(d_i, d_j)$ est une mesure de similarité

De **(2.11)** et **(2.12),** on a **:**

$$Q_S = C_S \bullet C_S = \|C_S\|^2 \qquad \textbf{(2.14)}$$

4.4. Difficultés particulières dans le clustering documentaire

La rédaction en langage naturel des documents à classer, à l'opposé du langage informatique, provoque généralement des difficultés au clustering documentaire, notamment :

La redondance : le langage naturel est un langage équivoque, c'est-à-dire, il y a plusieurs façons d'exprimer la même idée.

L'ambiguïté : ce qui est exprimé, dans un texte en langage naturel, possède souvent plusieurs interprétations.

L'implicite : tout n'est pas exprimé dans un texte

S'ajoute à ces particularités la grande dimensionnalité des descripteurs, et la subjectivité de la décision prise par les experts qui évaluent le résultat du clustering.

4.5. Travaux connexes en clustering documentaire en langue arabe

Au cours de notre travail, nous avons recensé deux principaux travaux en clustering des corpus en langue arabe.

Sur une analyse morphologique basée sur les connaissances linguistiques et en utilisant les n-gramm, les auteurs [Sawaf et al., 2001] ont employé une approche statistique

(basée sur la technique de la maximisation de l'entropie) pour le clustering d'une base d'articles arabes couvrant plusieurs domaines tels que la politiques, l'économie, etc.

[Huot et Coupet, 2005] ont développé un algorithme (intégré dans le logiciel standard *Insight Discoverer Clusterer* de TEMIS) qui, à partir de descripteurs en arabe, regroupe les documents similaires dans des classes en fonction de leurs ressemblance sémantique et de leur proximité thématique.

5. Conclusion

Les techniques de la classification non supervisée étant nombreuses, nous avons présenté, dans la première partie de ce chapitre, les principales techniques qui sont généralement utilisées dans le clustering. Nous avons commencé par voir ce qui est un clustering, les étapes d'un processus de clustering ainsi que les mesures de similarités généralement utilisées, nous avons vu aussi, les différentes techniques de l'évaluation de la qualité des clusters générés. Ensuite, nous sommes passés à la présentation de deux grandes familles du clustering notamment les méthodes dites hiérarchiques et les méthodes dites méthodes avec partitionnement. Nous nous sommes attardés sur les différentes techniques de ces deux familles ainsi que leurs variantes et surtout sur les deux techniques que nous avons utilisées dans notre approche et qui sont la méthode hiérarchique par agglomération (ascendante) et la méthode des k-médoïds et sa variante PAM *(Partitioning Around Medoids)*.

Dans une deuxième partie consacrée essentiellement au clustering documentaire ainsi que ses domaines d'application, nous nous sommes intéressé plutôt à la recherche documentaire. Nous avons vu ce qui un processus de recherche documentaire ainsi que les techniques de son évaluation telles que la mesure de précision et la mesure du rappel et ceci vu l'importance de cette recherche dans notre travail.

Enfin, nous avons traités les mesures utilisées pour l'évaluation d'un clustering documentaire ainsi que les difficultés qui peuvent en découler.

Chapitre III
La langue arabe

1. Introduction

La langue arabe est une langue sémitique (chamito-sémitique dans [Leclerc, 2000]**)** avec une morphologie très riche **[**Diab et al., 2004**]**. Elle est considérée comme une langue difficile à maîtriser dans le domaine du traitement automatique de la langue en égard à ses propriétés morphologiques et syntaxiques [Aljlayl et Frieder, 2002**],** [Larkey et al., 2002], [Larkey et al., 2005]**.** A partir du 7eme siècle, et grâce à la propagation de l'islam et la diffusion du Coran, l'arabe s'est expansé vers les quatre coins du monde [Leclerc, 2000]. Cette propagation a fait que plusieurs dialectes arabes se sont développés à l'intérieur de différentes communautés. Ces dialectes ne sont pas mutuellement compréhensibles ce qui a donné lieu à la naissance de l'Arabe Modern Standard (*Modern Standard Arabic, MSA*) qui est une forme simplifiée de l'arabe classique [Khoja, 2001]. Le MSA n'utilise pas les formes compliquées que l'on retrouve dans l'arabe classique, mais il emploie plutôt, ce que nous qualifions de formes légères. Le MSA est utilisé par les institutions académiques, les médias, les recherches, etc.

Actuellement les travaux de recherches continuent à aborder des problématiques tels que la recherche d'information, la syntaxe, la traduction automatique, le résumé automatique, etc. [Abdelali et al., 2004a]. Ces travaux se sont focalisés surtout sur l'aspect morphologique qui est, quant à lui, une véritable source d'ambiguïté.

> *A la différence des autres langues comme le français ou l'anglais, dont les étiquettes grammaticales proviennent d'une approche distributionnelle caractérisée par une volonté "d'écarter toute considération relative au sens", les étiquettes de l'arabe viennent d'une approche où le sémantique côtoie le formel lié à la morphologie du mot, sans référence à la position de ce dernier dans la phrase*. [Débili et al., 2002].

Cette approche est matérialisée par les notions de schèmes et de fonctions qui occupent une place importante dans la grammaire de l'arabe [Douzidia, 2004]. Par exemple, le mot français **ferme**, est hors contexte, un substantif, un adjectif ou un verbe. Alors que le mot

arabe RaLaKa غَلَقَ est un verbe à la 3ème personne masculin singulier de l'accompli actif, par contre sa forme non voyellée ou non vocalisée غلق (dans l'exemple donné ne sont représentées que les consonnes RLK) admet quatre catégories grammaticales :

Substantif masculin singulier (RaLKun غَلَقٌ : une fermeture),

Verbe à la troisième personne masculin singulier de l'accompli actif (RaLaKa غَلَقَ: il a fermé ou RaLLaKa غَلَّقَ : il a fait fermé),

Verbe à la troisième personne masculin singulier de l'accompli passif (RuLiKa غُلِقَ: il a été fermé),

Verbe à l'impératif deuxième personne masculin singulier (RaLLiK غَلِّقْ : fais fermer).

De ceci, on constate aussi l'étendue du rôle que jouent les voyelles dans les mots arabes, non seulement parce qu'elles enlèvent l'ambiguïté, mais aussi parce qu'elles donnent l'étiquette grammaticale d'un mot indépendamment de sa position dans la phrase. Un autre rôle que les voyelles peuvent jouer est celui des accents en français. Par exemple, un mot comme **peche** peut être interprété comme **pêche**, **pèche** et **péché**. En arabe un mot comme **جمل** peut être interprété comme **جَمَل** « chameau» ou **جُمَل** « phrases» ou encore **جَمَّل** « rendre beau ou garnir ». En plus, en arabe chaque lettre de chaque mot devrait posséder sa voyelle ce qui n'est en général pas le cas.

2. Particularité de la langue arabe

L'alphabet de la langue arabe compte 28 lettres (Tableau 3.1) qui changent de forme et de présentation selon leurs positions [Boulaknadel, 2005], ce qui les rendent extensibles à 90 lettres avec l'ajout des voyelles [Aljlayl et Frieder, 2002]. Outre l'orientation droite-gauche, la langue arabe, langue sémitique, et par opposition aux langues indo-européennes, possède des caractéristiques telles que l'agglutination, la structure particulière combinant schème et radical et la non vocalisation. En plus ses lettres changent de forme de présentation selon leur position (au début, au milieu ou à la fin du mot). Le tableau 3.2 montre les variations de la lettre ك (kef). Toutes les lettres se lient entre elles sauf (ذ, د, ز, ر, و, ا) qui ne se joignent pas à gauche.

Les voyelles, quant à elles, sont de deux types. Les voyelles courtes ou brèves qui ont la forme d'une marque diacritique (ـَـِـُـْ) placée au dessus ou au dessous des lettres, tandis que les voyelles longues (ا،و،ي) collent aux consonnes et sont toujours écrites, même dans les

formes non vocalisées. Le *tanwiin* (ـًـٍـٌ) est un autre genre de voyelles [El kassas, 2005], il marque l'indéfini et est réalisé par un signe diacritique fusionné au signe de la voyelle courte.

Lettre arabe	Correspondant français	Prononciation
ا	a	Alef
ب	b	Ba'
ت	t	Ta'
ث	Th	Tha'
ج	j	Jim
ح	h	Hha'
خ	Kh	Kha'
د	d	Dal
ذ	d	Thal
ر	r	Ra
ز	z	Zayn
س	s	Sin
ش	Sh	Shin
ص	S	Sad

Lettre arabe	Correspondant français	prononciation
ض	D	Dad
ط	T	Tah
ظ	Z	Thah
ع	' '	Ayn
غ	gh	Ghayn
ف	f	Fa
ق	q	Qaf
ك	k	Kaf
ل	l	Lam
م	m	Mim
ن	n	Nun
ه	h	Ha
و	w	Waw
ي	y	Ya

Tableau 3.1 : Les 28 lettres de l'alphabet arabe [Leclerc, 2000]

A la fin d'une lettre non joignable	A la fin	Au milieu	Au début
ك	ـك	ـكـ	كـ

Tableau 3.2 : Variation de la lettre ك kef

Les voyelles sont nécessaires à la lecture et à la compréhension correcte d'un texte, elles permettent de différencier des mots ayant la même représentation (Tableau 3.3). Cependant, elles ne sont utilisées (ici on parle des voyelles courtes) que dans des contextes spéciaux tels que les livres didactiques, les dictionnaires ou le Coran [Larkey et al., 2002], [Larkey et al., 2005].

Les textes courants rencontrés dans les journaux et les livres n'en comportent habituellement pas. En plus certaines lettres comme ا (Alef) peuvent symboliser le آ ,أ ou إ; de même que pour les lettres ى et ه qui symbolisent respectivement ي et ة [Xu et al., 2002].

Mot sans voyelle	1ère interprétation		2ème interprétation		3eme interprétation	
علم	عَلَمَ	Il a enseigné	عِلْمٌ عَلَمٌ	Science Drapeau	عُلِّمَ عُلِمَ	Il a été enseigné Il a été su

Tableau 3.3 : Ambiguïté causée par l'absence des voyelles dans le mot علم

Il faut noter que l'ambiguïté causée par l'absence des voyelles courtes est atténuée par l'association de formes, de sens, de contexte, etc.

2.1. Morphologie arabe

Le lexique arabe est basé sur trois catégories de mots; les verbes, les noms et les particules. Ces catégories sont dérivées de quelques milliers de racines (10000) [Darwish, 2002], [Larkey et al., 2005]. Les racines des verbes et des noms sont souvent à trois consonnes radicales ou trilitères, et avec un degré moindre à quatre consonnes et rarement à cinq [Darwish, 2002], [Tuerlinckx, 2004]. Un même concept sémantique peut engendrer une famille de mots à l'aide de différents schèmes. Ce phénomène est une caractéristique à 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 (environ 150 [Douzidia, 2004], [CIEP, 2007]) qui sont des adjonctions et des manipulations de la racine (Tableau 3.4). On peut ainsi dériver un grand nombre de noms, de formes et de temps verbaux.

Schèmes	**RLK**	**غلق**	**Notion de fermer**		**MSK**	**مسك**	**Notion de tenir**
$R_1aR_2aR_3a$	RaLaKa	غَلَقَ	Il a fermé		MaṢaKa	مَسَكَ	Il a tenu
$R_1âR_2iR_3$	RâLik	غَالِق	Fermant		MâSiK	مَاسِك	Teneur
$maR_1R_2uR_3$	maRLuK	مَغْلُوق	Fermé		maMSuK	مَمْسُوك	Tenu
$R_1uR_2iR_3a$	RuLiKa	غُلِقَ	Il a été fermé		MuSiKa	مُسِك	Il a été tenu
…							

Tableau 3.4 : Exemple de schèmes pour les mots غلق et مسك

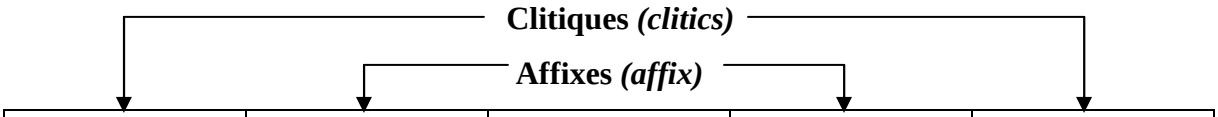
Les lettres en majuscule (R_i) désignent les consonnes de base qui composent la racine, les voyelles (â, a, i,..) désignent les voyelles, enfin les consonnes en minuscule (m,..) sont des consonnes de dérivation utilisées dans les schèmes.

2.2. Structure de mot arabe

Le mot ou l'unité graphique, suite de graphèmes entre deux blancs, correspond le plus souvent en langue arabe non pas à une forme ou «unité susceptible de figurer sous une entrée lexicale ou lemme», mais à une suite de formes collées les unes aux autres [Tuerlinckx, 2004]. Les mots du texte sont des formes agglutinées.

Un mot arabe peut représenter une phrase en langue latine [Douzidia, 2004],[Diab et al., 2004], sa structure peut être composée d'un corps schématique (stem), des antéfixes ou proclitiques (*proclitics*) qui sont des prépositions ou des conjonctions, des affixes (*affix*) (préfixes et suffixes) qui expriment les traits grammaticaux et indiquent les fonctions comme le cas du nom, mode du verbe et les modalités (nombre, genre, personne,…) et des postfixes ou enclitiques (*enclitics*) qui sont des pronoms personnels.

Par exemple le mot **أَتَأْكُلُونَهَا,** qui veut dire : « Est-ce que vous la mangez ?», se décompose selon le tableau 3.5.



Postfixe	Suffixe	Corps schématique	Préfixe	Antéfixe
هَا	ونَ	أْكُلُ	تَ	أ
Pronom suffixe complément du nom	Suffixe verbal exprimant le pluriel	verbe	Préfixe verbal du temps de l'inaccompli	Conjonction d'interrogation

Tableau 3.5 : Décomposition d'un mot arabe

2.3. Catégories des mots

L'arabe considère 3 catégories de mots [Khoja, 2001], [Maamouri et Bies, 2004], verbe nom et particules. Certains grammairiens ajoutent une catégorie instruments ou articles recoupant plus ou moins celle des particules [Tuerlinckx, 2004], alors que d'autres donnent une toute autre catégorisation [El Kassas, 2005].

2.3.1. Verbe

Un verbe est une entité exprimant un sens dépendant du temps, c'est un élément fondamental auquel se rattachent directement ou indirectement les divers mots qui constituent l'ensemble. La catégorie du verbe contient toutes les unités lexicales référant à un état ou à une action au passé, au présent ou au futur [El kassas, 2005].

La plupart des mots en arabe, dérivent d'un verbe de trois lettres [Douzidia, 2004]. Chaque verbe est donc la racine d'une famille de mots. Comme en français, le mot en arabe se déduit de la racine en rajoutant des suffixes et/ou des préfixes. La conjugaison des verbes dépend de plusieurs facteurs. Le temps (accompli, inaccompli, impératif), le nombre du sujet (singulier, duel, pluriel), le genre du sujet (masculin, féminin), la personne (première, deuxième, troisième) et le mode (actif, passif).

Exemple :

Les trois lettres غ+ل+ق (R+L+K) donnent le verbe غلق RaLaKa (fermer). Dans tous les mots qui dérivent de cette racine, on retrouvera ces trois lettres (Tableau 3.4).

Conjugaison du verbe

Pour conjuguer les verbes, on ajoute des préfixes et des suffixes. La langue arabe dispose de trois temps [El kassas, 2005] :

*** L'accompli** : correspond au passé et se distingue par des suffixes. Par exemple pour le pluriel féminin on a : غلقن RalaKna (elles ont fermé), pour le pluriel masculin on a : غلقوا RaLaKuu (Ils ont fermé).

*** L'inaccompli**

Présent : présente l'action en cours d'accomplissement, ses éléments sont préfixés. Par exemple pour la troisième personne du singulier on a يغلق yaRLiKu (il ferme) pour le masculin et تغلق taRLiKu (elle ferme) pour le féminin.

Futur : correspond à une action qui se déroulera au futur et est marqué par l'antéposition de س (sa) collée au début du verbe, ou سوف (sawfa) positionnée avant le verbe. Par exemple on a سيغلق sayaRliku (il fermera) ou يغلق سوف sawfa yaRLiKu (il va fermer).

*** L'impératif**

Présente l'action directive pour l'accomplissement. Par exemple pour la deuxième personne du singulier on a : اغلق aRLiK (ferme) pour le masculin et اغلقي aRLiKi (ferme) pour le féminin.

2.3.2. Nom

Le nom est l'élément désignant un être ou un objet qui exprime un sens indépendant du temps. La catégorie des noms regroupe toutes les unités lexicales référant à un sens qui n'est pas lié au temps. Cette catégorie comprend le substantif et l'adjectif (الصفة والموصوف) [El kassas, 2005]. Les substantifs sont de deux catégories, ceux qui sont dérivés de la racine verbale et ceux qui ne le sont pas comme les noms propres et les noms communs. La déclinaison des noms se fait selon les règles suivantes :

Le féminin singulier : On ajoute le ة. Par exemple جالس (assis) devient جالسة (assise)

Le féminin pluriel : De la même manière, on rajoute pour le pluriel les deux lettres ات. Par exemple جالس (assis) devient جالسات (assises).

Le masculin pluriel : Pour le pluriel masculin, les deux lettres ين ou ون sont rajoutées dépendamment de la position du mot dans la phrase (sujet ou complément d'objet). Par exemple : جالس devient جالسين ou جالسون (assis).

Le Pluriel irrégulier : Il suit une diversité de règles complexes et dépend du nom. Par exemple : طفل (enfant) devient أطفال (enfants) ou encore صديق (ami) devient أصدقاء(amis).

Le phénomène du pluriel irrégulier en langue arabe est très fréquent, il pose un défi à la morphologie, non seulement à cause de sa nature non concaténative, mais aussi parce que son analyse dépend fortement de la structure comme pour les verbes irréguliers [Kiraz, 1996].

Il faut noter que cette situation est retrouvée en langue anglaise mais avec un nombre réduit de noms et un petit nombre de verbe très fréquents [Larkey et al., 2002].

Certains dérivés nominaux associent une fonction au nom : l'Agent (celui qui fait l'action), l'Objet (celui qui a subit l'action), l'Instrument (désignant l'instrument de l'action) et le Lieu.

Pour les pronoms personnels, le sujet est inclus dans le verbe conjugué. Il n'est donc pas nécessaire (comme c'est le cas en français) de précéder le verbe conjugué par son pronom. On distinguera entre singulier, duel (deux) et pluriel (plus de deux) ainsi qu'entre le masculin et féminin.

Exemple:

Pour le verbe غلق on aura :

غلقت pour le singulier

غلقتما pour le duel

غلقتم ou غلقتن pour le pluriel

2.3.3. Particules

Ce sont des entités qui sont principalement les mots outils comme les conjonctions de coordination et de subordination. Les particules sont classées selon leurs sémantiques et leurs fonctions dans la phrase, on en distingue plusieurs types (introduction, explication, conséquence, …). Elles jouent un rôle important dans l'interprétation de la phrase [Douzidia, 2004]. Elles servent à situer des événements (faits) ou des objets par rapport au temps ou au lieu (espace). Elles permettent également l'enchaînement et la cohérence dans le texte.

On a, par exemple, des particules qui désignent un temps comme بعد ، قبل ، منذ (après, avant, pendant), un lieu حيت (où), ou qui sont de référence الذين١ (ceux)….

Il est à noter que les particules peuvent avoir aussi des préfixes et suffixes ce qui rajoute une complexité quant à leur identification, on a par exemple بعدك ، قبلك (après toi, avant toi) et كالذين (pareil à ceux).

3. Problèmes posés au traitement automatique de la langue arabe

Comme il a été mentionné auparavant, l'un des aspects les plus complexes de la langue arabe est l'absence des voyelles dans le texte, ce qui risque de générer une certaine ambiguïté au niveau du sens d'un mot et de sa fonction, surtout lorsqu'un mot peut être dérivé de plusieurs racines différentes [Attia, 2000]. De ceci, plusieurs problèmes surgissent, dans notre

cas, nous nous limitons au problème de la détection de la racine, puisque les autres problèmes relèvent du traitement automatique du langage naturel (TALN). Néanmoins, nous exposons une deuxième difficulté qui est l'agglutination.

3.1. Détection de racine

Les racines véhiculent une idée principale, par exemple, **كتب** «k-t-b» indique l'idée d'écriture. Lorsque l'on accole les éléments flexionnels, avant, entre ou après les lettres de base, cela nous donne un ensemble de mots associés (*surface forms*) [Aljlayl et Frieder, 2002], **كَتَبَ** «écrire», mais aussi **كِتَاب** «livre», **مَكتب** «bureau», **مكتبة** «bibliothèque» et **كاتب** «auteurs». Dans le chemin inverse, le problème de détection de la racine est relativement simple, il suffit de connaître le schème par lequel il a été dérivé et supprimer les éléments flexionnels (antéfixes, préfixes, suffixes, postfixes) qui ont été ajoutés.

K. Darwish et *D. W. Oard* (2002) ont proposé une liste de préfixes et de suffixes (Tableau 3.6), qui ont été déterminés par un calcul de fréquence sur une collection d'articles arabes de l'*Agence France Press* (AFP) [Darwish et Oard, 2002]. Plusieurs éléments de cette liste ont été utilisés dans certains travaux pour la radicalisation de mots arabes tels que le stemmer *Al-Stem* de *K. Darwish*.

Préfixes							
فا	ويـ	الـ	ومـ	نتـ	متـ	بتـ	والـ
لا	فيـ	للـ	كمـ	بمـ	وتـ	يتـ	فالـ
با	وا	ليـ	فمـ	لمـ	ستـ	لتـ	بالـ
Suffixes							
ا	ـة	ين	ية	هم	ته	وه	ات
	ـه	يه	تك	هن	تم	ـان	وا
	ي	ية	نا	ها	كم	تي	ون

Tableau 3.6 : Liste des préfixes et suffixes les plus fréquents *(Al-stem)*

Afin de détecter la racine d'un mot en arabe, une analyse morphologique doit tenter d'identifier et de séparer les différentes combinaisons de préfixes et de suffixes possibles et semblables aux mots préfixés comme les conjonctions wa- و et fa- ف, des prépositions préfixées comme bi- ب et li- ل, l'article défini ال , des suffixes de pronom possessif. Ces combinaisons sont retrouvés en enlevant progressivement des préfixes et des suffixes et en essayant de faire correspondre toutes les racines produites à un schème afin de retrouver la racine [Darwish, 2002].

Dans le cas où un mot peut être dérivé de plusieurs racines différentes, la détection de la racine est plus difficile, en particulier en absence de voyelles [Attia, 2000].

Par exemple (Tableau 3.7), pour le mot arabe ايمان AymAn, les préfixes/antéfixes possibles sont: "Ø","A ا" et "Ay اي " et les suffixes/postfixes possibles sont : " Ø " et "An ان ", sans compter que ce mot peut aussi représenter un nom propre إيمَان (Imène).

Stem	Préfixes/ antéfixes	schème	suffixes/ postfixes	Racine	Signification
AyMaN ايمان	Ø	R1yR2aR3	Ø	AMN امن	Croyance
YMaN يمان	A ا	R1R2aR3	Ø	YMN يمن	Convenant
MAN مان	Ay اي	R1R2R3	Ø	MAN مان	Va-t-il approvisionner
AYM ايم	Ø	R1R2R3	An ان	AYM ايم	Deux veuves

Tableau 3.7 : Les stems possibles pour le mot ايمان

Certains verbes sont considérés comme irréguliers, ce sont ceux qui portent des consonnes particulières dites faibles (ي ,ا ,و) [Douzidia, 2004]. Ils sont appelés ainsi parce que, lors de leurs déclinaisons, chacune de ces lettres particulières est conservée, remplacée ou éliminée.

Exemple :

Le tableau 3.8 donne les différentes déclinaisons du verbe قال (Il a dit) :

Caractère ا est remplacé par	قال	Dire
ا	قال	Il a dit
و	يقول	Il dit
يـ	قيل	Il a été dit
Ø	قل	Dis

Tableau 3.8 : Exemple de déclinaisons du verbe irrégulier قال dire

3.2. Agglutination

L'agglutination par laquelle les composantes du mot sont liées les unes aux autres représente une autre difficulté en traitement automatique de la langue arabe. Contrairement aux langues latines, l'arabe est une langue agglutinante; les articles, les prépositions et les pronoms collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent [Larkey et al., 2002]. Ceci engendre une ambiguïté morphologique au cours de l'analyse des mots [Boulaknadel, 2005], ce qui complique la tache de l'analyse morphosyntaxique pour identifier les vrais composants du mot.

Par exemple, le mot أَلَمُهُمْ ALaMuhum (leur douleur) dans sa forme voyellée n'accepte qu'une seule segmentation : أَلَمُ + هُمْ (ALaMu+hum). Dans sa forme non voyellée المهم (ALMHM), le même mot accepte au moins les trois segmentations données dans le tableau 3.9.

Segmentation possible		Traduction en français
أ + لم + هم	a + LM + hm	Les a-t-il ramassé
ألم + هم	ALM + hm	Leur douleur
	ALM + hm	Il les a fait souffrir
أل + مهم	Al + MHM	L'important

Tableau 3.9 : Exemple de segmentation du mot المهم

L'amplification de l'ambiguïté dans la segmentation s'opère selon deux façons [Débili et al., 2002]:

- d'abord, il y a plus d'unités ambiguës dans un texte non voyellé que dans son correspondant voyellé, mais aussi, les unités ambiguës acceptent plus de segmentations dans le texte non voyellé.

- de plus le fait de procéder à la radicalisation par la troncature des préfixes avant les suffixes (et réciproquement) peut influencer les résultats.

En reconsidérant l'exemple du tableau 3.9, sur un texte où la notion de douleur est importante, le fait d'avancer la suppression des préfixes avant les suffixes les mots comme المهم (leur douleur pour le pluriel), المهما (leur douleur pour le duel) exprimeront une toute autre notion.

4. Conclusion

Dans ce chapitre, dédié aux particularités de la langue arabe, nous avons essayé de montrer le caractère productif, dérivationnel et flexionnel de cette langue où chaque mot peut avoir un grand nombre de formes d'une part, et d'autre part de sens, qui est du aux ambiguïtés causées, essentiellement, par l'absence de voyelles et l'agglutination des mots par rapport à d'autres langues comme le français ou l'anglais, ce qui rend la tâche d'un clustering documentaire très compliquée.

Nous avons commencé par décrire la morphologie de cette langue, suivi de la structure de ses mots ainsi que leurs catégories qui sont essentiellement, le verbe, le nom et les particules.

Les problèmes qui pourraient être posés par cette langue au traitement automatique, ont été étudiés. Nous nous sommes limités, exclusivement, au problème de la détection de la racine, bien qu'elle soit difficile pour les langues avec des morphologies complexes comme l'arabe, elle est particulièrement importante et utile et spécialement dans un système de recherche documentaire, système que nous avons utilisé pour tester la validité de notre approche qui sera l'objet du quatrième chapitre.

Chapitre IV
Classification non supervisée pour la recherche documentaire

1. Introduction

L'ampleur de notre sujet, nous a obligé à suivre une méthodologie de travail qui consiste à tester l'influence de plusieurs facteurs importants tels que le stemming, les méthodes de classification non supervisée utilisées et le nombre de clusters choisis, avant d'attaquer le principal objectif qui est l'influence de la classification non supervisée dans une recherche documentaire. Les travaux [Kelaiaia et al., 2007a], [Kelaiaia et al., 2007b] rentrent dans ce cadre et visent à mesurer l'influence mutuelle entre ces facteurs et les caractéristiques spécifiques de la langue arabe. Dans ce qui suit nous exposant les différentes étapes par lesquelles notre étude est passée.

Nous avons testé l'influence d'un prétraitement tel que le stemming sur la qualité des clusters obtenus par deux méthodes de classification non supervisée, hiérarchique ascendante et par partitionnement, dans une première évaluation.

L'influence de ce prétraitement sur la précision d'un système de recherche classique, en réponse à un jeu de questions composé de 21 requêtes (exposées en annexe 1), a été testée dans une deuxième évaluation.

Dans une troisième évaluation, qui est l'objet de l'approche proposée, nous avons mesuré l'influence d'une classification non supervisée (clustering), en utilisant une méthode hiérarchique ascendante et une méthode par partitionnement, d'un corpus de textes arabes sur la qualité d'une recherche documentaire 'locale', en premier lieu sur des textes ayant subis des prétraitements, et en second, sur des textes à l'état brut *(raw text)*. L'évaluation s'est faite par rapport à une recherche documentaire effectuée à l'aide d'un moteur de recherche classique locale, toujours en réponse au même jeu de requêtes.

Le but de l'utilisation de deux méthodes de classification non supervisée est de permettre de mettre en évidence l'influence de chacune des deux méthodes sur la qualité

d'une recherche documentaire. Cette étude est menée sur plusieurs nombres de clusters, ce qui a permis aussi de mesurer la qualité de la classification obtenue et l'influence du nombre de clusters sur la précision de la recherche documentaire.

Les mesures d'évaluation choisies étant la précision et le rappel, sont déterminées par rapport aux résultats obtenus par des humains, travail qui est, quant à lui, décrit dans la section suivante.

Dans ce qui ce suit nous allons exposer les différents travaux qui nous ont mené aux résultats discutés dans le prochain chapitre.

2. Stratégie de l'étude menée

La stratégie que nous avons adoptée, s'articule sur trois phases essentielles, la première étant la préparation des données, consiste à préparer le paramètre autour duquel nous avons construit notre évaluation, à savoir, la pertinence de chaque document par rapport à une requête, ainsi que les prétraitement que doivent subir les textes du corpus. La deuxième phase a été consacrée au processus de clustering porté sur le corpus pour la génération des différentes partitions (5, 7, 9, 11, 13 et 15 clusters) avec les deux méthodes choisies et la troisième phase a été destinée essentiellement à l'approche proposée.

2.1. Phase 1 : Préparation des données

Au cours de cette phase, nous avons mené deux travaux en parallèle.

En premier lieu et afin de permettre, par la suite, de mesurer la qualité des listes renvoyées par une recherche documentaire avec l'approche proposée par rapport à celle effectuée par un moteur de recherche classique locale, en réponse à une requête, une troisième liste est établie manuellement par des humains. Pour ce faire, nous avons soumis l'ensemble du corpus, décrit dans la section 2, chapitre 5, à un groupe d'étudiants qui ont pris le soin de lire touts les documents et de déterminer ceux qui sont pertinents pour chacune des 21 requêtes (écrites elles aussi par les membres du même groupe).

En même temps, nous avons procédé à la préparation de l'ensemble du corpus pour les différentes expérimentations. Nous avons suivi les étapes suivantes :

1. Transformation automatique des textes écrits sous XML vers une forme textuelle
2. Nettoyage de textes, Translittération et Tokenisation
3. Stemming
4. Indexation en vecteur TF-IDF

2.2. Phase 2 : Clustering documentaire

Le but de cette phase est de préparer les différentes partitions afin de les utiliser dans la troisième phase. Pour ce faire, nous avons décidé d'utiliser une méthode hiérarchique et une méthode par partitionnement, étant donné que ces deux dernières sont les plus répondues en classification non supervisée documentaire [Jardino, 2004]; En ce qui concerne l'approche hiérarchique, nous avons opté pour la méthode d'agglomération ascendante avec lien complet (*complete link*) décrite dans la section 3.3.1.1, chapitre 2, par l'algorithme 2.1. Pour l'approche par partitionnement, nous avons opté pour la méthode des K-médoïds et son implémentation PAM *(Partitioning Around Medoids)* décrite dans la section 3.3.2.2, chapitre 2, par l'algorithme 2.4. Le choix de PAM est justifié par le fait que dans le cas contraire (exemple choix d'une méthode de type centre mobile), nous pouvons tomber sur un représentant de cluster qui n'est un des documents à classer. L'implémentation de ces deux méthodes est décrite dans la section 4.2.1, chapitre 5.

A la fin de cette phase nous devions avoir, au total, 24 partitions présentées comme suit :

- Méthode d'agglomération ascendante appliquée au corpus n'ayant pas subi un traitement de stemming : 6 partitions (5, 7, 9, 11, 13, 15)
- Méthode d'agglomération ascendante appliquée au corpus ayant subi un traitement de stemming : 6 partitions (5, 7, 9, 11, 13, 15)
- Méthode des K-médoïds appliquée au corpus n'ayant pas subi un traitement de stemming : 6 partitions (5, 7, 9, 11, 13, 15)
- Méthode des K-médoïds appliquée au corpus ayant subi un traitement de stemming : 6 partitions (5, 7, 9, 11, 13, 15)

2.3. Phase 3 : Méthodologie de l'approche proposée

Au cours de cette phase une réponse devait être donnée à chacune des deux questions suivantes :

Comment traiter une requête en langage naturel ?

Comment la liste des documents retournée, sera-t-elle constituée ?

2.3.1. Traitement d'une requête

Dans notre travail, une requête est prise à l'état brut, c'est-à-dire, que nous ne faisons aucun traitement d'enrichissement des requêtes (*query expansion*) qui se fait, généralement, par l'ajout automatique de termes grâce à l'utilisation d'un thesaurus.

Chaque requête introduite doit subir le même prétraitement que les textes du corpus (nettoyage de textes, translittération, tokenisation, stemming et représentation en vecteur TF-IDF).

2.3.2. Constitution de la liste des documents retournée

La liste des documents renvoyée, en réponse à la requête introduite, est construite à partir des documents du cluster ayant le représentant le plus proche de la requête. Il faut déterminer le représentant du cluster le plus proche de la requête. Ici les textes sont ordonnés selon leurs distances avec ce dernier. Ensuite la liste est complétée par les documents du cluster le plus proche du clusters gagnant avec le même critère d'ordonnancement, et ainsi de suite. Ce qui traduit par l'algorithme 4.1.

1. faire subir à la requête le même prétraitement qu'un texte;
2. éffectuer un calcul de distance entre l'ensemble formé par les représentants des clusters et la requête : le représentant le plus proche de la requête gagne;
3. **l**a liste est constituée dans l'ordre d'apparition des documents dans le cluster du représentant gagnant;
4. la liste est complétée par les documents des clusters restants selon l'ordre de distances de leurs représentants du cluster gagnant.

Algorithme 4.1 : Construction de la liste des documents retournés en réponse à une requête

Il faut noter que le traitement de stemming est inhibé dans le cas où nous avons testé l'approche proposée sur le corpus n'ayant pas subi de stemming.

Notant aussi que la distance employée pour mesurer la distance entre deux textes dans la phase 2 et entre un texte et une requête dans la phase 3, est la distance du cosinus déjà décrite dans la section 4.2, chapitre 2, équation 2.11.

3. Critères d'évaluation

3.1. Détermination des documents pertinents pour chaque requête

Comme il a été mentionné dans le paragraphe 2.1, nous avons confié le soin de la détermination de la pertinence d'un document par rapport à une requête à groupe d'étudiants. En effet, l'expertise humaine dans ce domaine n'est plus à démonter, cette pratique est exercée dans les plus grandes compagnes telles que TREC, Amaryllis, DEFT... etc.

3.2. Evaluation de la recherche effectuée

Les deux mesures d'évaluation que nous avons utilisées sont la précision et le rappel, deux mesures qui sont largement utilisées dans ce domaine [Rijsbergen, 1979]. Elles ont été détaillées dans la section 4.1.4, chapitre 2.

4. Conclusion

Ce chapitre se voulait une présentation de la démarche que nous avons suivi afin de tester une approche basée sur le clustering documentaire pour l'amélioration d'une recherche documentaire locale.

Dans cette démarche et avant d'exposer le principe ainsi que les différentes étapes de cette approche, nous avons mis la lumière sur les différentes phases de la préparation des textes. Cette préparation nous a permis d'effectuer plusieurs analyses et évaluations et d'obtenir deux résultats intermédiaires très intéressants, qui sont l'influence du stemming sur la qualité d'un processus de clustering ainsi que sur une recherche documentaire locale classique.

Nous nous sommes tournés ensuite, vers la présentation du fonctionnement de notre approche en commençant par donner la manière dont une requête est traitée en langage naturel puis le mécanisme de la constitution de la liste des documents retournée en réponse à cette requête.

Pour l'évaluation des résultats obtenus par notre approche et ceux obtenus par une recherche classique, nous nous sommes alignés aux grandes compagnes internationales pour le jugement de la pertinence d'un document par rapport à une requête, à savoir, l'expert humain. Suite aux résultas obtenus, nous avons calculé les mesures les plus renommées dans ce domaine qui sont la précision et le rappel.

Le déroulement et les résultats obtenus par cette démarche sont exposés dans le chapitre suivant.

Chapitre V
Expérimentations, résultats et discussion

1. Introduction

Ce chapitre traite les résultas obtenus au fur et à mesure du déroulement des différentes expérimentations. Nous allons commencer par donner une présentation du corpus sur lequel ont été menés nos travaux, l'outil que nous avons développé ainsi que les différents outils utilisés.

2. Présentation du corpus

La collection des manuscrits, des livres et des articles de presses pour les analyser est une tache très ardue dans sa nature [Abdelali et al., 2004b]. Grâce à l'avancée technologique dans le stockage informatique et l'accès à une large quantité d'information, la construction des corpus de textes continue à se développer.

Qu'est ce qu'un corpus ?

Un corpus est un ensemble de documents respectant deux critères, à savoir l'homogénéité; un corpus homogène couvre un domaine spécifique dans toute sa diversité et sa taille, représentée par le nombre de mots.

Pourquoi un corpus ?

La nécessité actuelle d'un corpus se veut dans les études en grammaire, lexicographie, variation du langage, linguistique historique, acquisition du langage et pédagogie du langage [Abdelali et al., 2004b].

Actuellement, nous pouvons trouver des corpus qui comprennent plusieurs centaines de millions de mots. Dès l'année 1989, L'agence de presse Reuters a mis à la disposition des chercheurs le corpus, le plus célèbre dans la communauté de la classification [Dunoyer, 2004], Reuters-21578, qui comporte 21578 dépêches financières, en langue anglaise, émises au cours de l'année 1987.

Pour ce qui est de langue arabe, le tableau 5.1 (source [Sulaiti, 2005][1]) représente les corpus disponibles 'selon des restrictions' sur des sources Web.

[1] D'autres sources peuvent être consultées tels que [NEMLAR, 2005] et [LDC, 2007]

Corpus	Source	Moyen	Taille	Utilisation	Origine
Buckwalter Arabic Corpus 1986-2003	Tim Buckwalter	Written	2.5 to 3 billion words	Lexicography	Public resources on the Web
Leuven Corpus (1990-2004)	Catholic University Leuven, Belgium	Written and spoken	3M words (spoken: 700000)	Arabic-Dutch /Dutch-Arabic learner's dictionary	Internet sources, radio & TV, primary school books
Arabic Newswire Corpus(1994)	University of Pennsylvania LDC	Written	80M words	Education and the development of technology	Agence France Presse, Xinhua News Agency, and Umma Press
CALLFRIEND Corpus (1995)	University of Pennsylvania LDC	Conversational	60 telephone conversations	Development of language identification technology	Egyptian native speakers
NijmegenCorpus (1996)	Nijmegen University	Written	Over 2M words	Arabic-Dutch / Dutch-Arabic dictionary	Magazines and fiction
CALLHOME Corpus (1997)	University of Pennsylvania LDC	Conversational	120 telephone conversations	Speech recognition produced from telephone lines	Egyptian native speakers
CLARA (1997)	Charles University, Prague	Written	50M words	Lexicographic purposes	Periodicals, books, internet sources from 1975-present
Egypt (1999)	John Hopkins University	Written	Unknown	MT	A parallel corpus of the Quran in English and Arabic
Broadcast News Speech (2000)	University of Pennsylvania LDC	Spoken	More than 110 broadcasts	Speech recognition	News broadcast from the radio of voice of America.
DINAR Corpus (2000)	Nijmegen Univ.,SOTETEL -IT, co-ordination of Lyon2 Univ	Written	10M words	Lexicography, general research, NLP	Unknown
An-Nahar Corpus (2001)	ELRA	Written	140M words	General research	An-Nahar newspaper (Lebanon)
Al-Hayat Corpus (2002)	ELRA	Written	18.6M words	Language Engineering and Information Retrieval	Al-Hayat newspaper (Lebanon)
Arabic Gigaword (2002)	University of Pennsylvania LDC	Written	Around 400M	Natual language processing, information retrieval, language modelling	Agence France Presse, Al-Hayat news agency, An-Nahar news agency, Xinhua news agency
E-A Parallel Corpus (2003)	University of Kuwait	Written	3M words	Teaching translation & lexicography	Publications from Kuwait National Council
General Scientific Arabic Corpus (2004)	UMIST, UK	Written	1.6M words	Investigating Arabic compounds	http://www.kisr.edu.kw/science/
Classical Arabic Corpus (CAC) (2004)	UMIST, UK	Written	5M words	Lexical analysis research	www.muhaddith.org and www.alwaraq.com
Multilingual Corpus 2004	UMIST, UK	Written	11.5M words (Arabic 2.5M)	Translation	IT-specialized websites-computer system and online software help-one book
SOTETEL Corpus	SOTETEL-IT,	Written	8M words	Lexicography	Literature, academic and

	Tunisia				journalistic material
Corpus of Contemporary Arabic (CCA) 2004	University of Leeds	Written and spoken	Around 1M words	TAFL	Websites and online magazines
DARPA Babylon Levantine Arabic Speech and Transcripts (2005)	University of Pennsylvania LDC	Spoken	About 2000 telephone calls	Machine translatioon, speech recognition & spoken dialogue system	Fisher style telephone speech collection

Tableau 5.1 : Corpus en langue arabe (source [Sulaiti, 2005])

La limitation de la disponibilité des corpus en langue arabe a incité plusieurs chercheurs à se retourner vers l'aspect construction des corpus. Plusieurs recherches sont apparues, nous citons ceux de [Goweder et De Roeck, 2001], [Darwish, 2002], [Maamouri et Bies, 2004], [Abdelali et al., 2004b] et [Abdelali et Cowie, 2004],.

Pour notre travail, toutes les expérimentations ont été portées sur le corpus CCA (*Corpus of Contemporary Arabic*) compilé par *Latifa El Sulaiti* [Sulaiti, 2003] sur Radio Qatar collectionné pour le projet de l'arabe contemporain en Juin 2003. Il comporte 402 textes sous XML partitionnés en 16 classes (tableau 5.2).

Classe	**Nbre de textes**	**Nbre de mots**	**Taille (M. Octet)**
Short stories	31	45.460	0.444
Education	10	25.574	0.295
Religion	19	111. 199	1.170
Autobiography	73	153.459	1.690
Sociology	30	85688	1.170
Tourist/travel	60	46.093	0.612
Recipes	9	4.973	0.070
Science	45	104.795	2.230
Sports	4	8.290	0.122
Economics	29	67.478	0.797
Children' stories	27	21.958	0.270
Health and medicine	32	40.480	0.464
Interviews	23	58.408	0.626
Politics	10	46.291	0.531
Total	**402**	**820.146**	**10.491**

Tableau 5.2 : Caractéristiques du corpus CCA

Il faut noter que dans notre étude, nous ne prenons pas en compte les classes décrites dans le tableau 5.2, puisque, elle est basée sur un processus de clustering. Ce processus suppose qu'il n y a aucune classification au préalable.

3. Présentation des environnements utilisés

3.1. Environnement YALE (*Yet Another Learning Environment*)

Développé par une équipe de chercheurs du département d'informatique de l'université de Dortmund, cet environnement est un Java open source qui est destiné au *Datamining* et au *Machine learning,* il est téléchargeable à partir du Web[1] ainsi que ses modules intégrables (*plugins*), source et installation, accompagnée d'une riche documentation.

Plusieurs versions de ce projet ont vu le jour mais sous une autre appellation, la plus récente est celle de *RapidMiner 4.1Beta* intégrant plus de 500 opérateurs répartis en catégories hiérarchiques. Les deux catégories prises en considération sont :

Catégorie des entrées/sorties contient les opérateurs pour la manipulation des attributs, la génération des sources d'exemples ainsi que les modèles, …

Catégorie des opérateurs d'apprentissage contient une multitude de classificateurs répartis en deux catégories supervisés et non supervisés

D'autres catégories comme le Prétraitement, le Posttraitement, la visualisations des résultats, les métas données, la validation des résultats, OLAP (*On-line Analytical Processing),* etc.

En plus de ces opérateurs, plusieurs modules (*plugins*) sont disponibles :

Module texte *(WVTool : Word Vector Tool)* : Pour la création des vecteurs de mot à partir de différentes sources (Collection, Chaîne de caractères, …).

Module *Value series* : Fourni des opérateurs pour l'extraction automatique d'attributs à partir des séries de données.

Module *Data Stream* : Destiné aux traitements des données types flux.

Module CRF (*Conditional Random Field*) : Ensemble d'opérateurs pour les tâches de classification séquentielle.

Une expérimentation sous *RapidMiner* est montée autour d'une Racine d'expérimentation (*Root Experiment*) (figure 5.1). Les opérateurs cités précédemment sont greffés au fur et à mesure du développement de l'expérimentation. Chaque opérateur dispose de quatre onglets, les deux plus importants sont : la configuration des paramètres de l'opérateur et son code XML.

[1] http://sourceforge.net/projects/yale/

3.2. Google Desktop

Pour pouvoir mesurer l'apport de l'approche proposée, nous avons comparé les résultats obtenus par cette dernière à ceux obtenus avec un moteur classique pour la recherche locale[1]. Le moteur que nous avons choisi est *Google Desktop*[2] version *5.5 beta* (figure 5.2).

Après son installation sur le micro-ordinateur, *Google Desktop* effectue une indexation (basée sur le *ranking*) de tous les fichiers, ce qui lui permet, par la suite, d'effecteur une recherche.

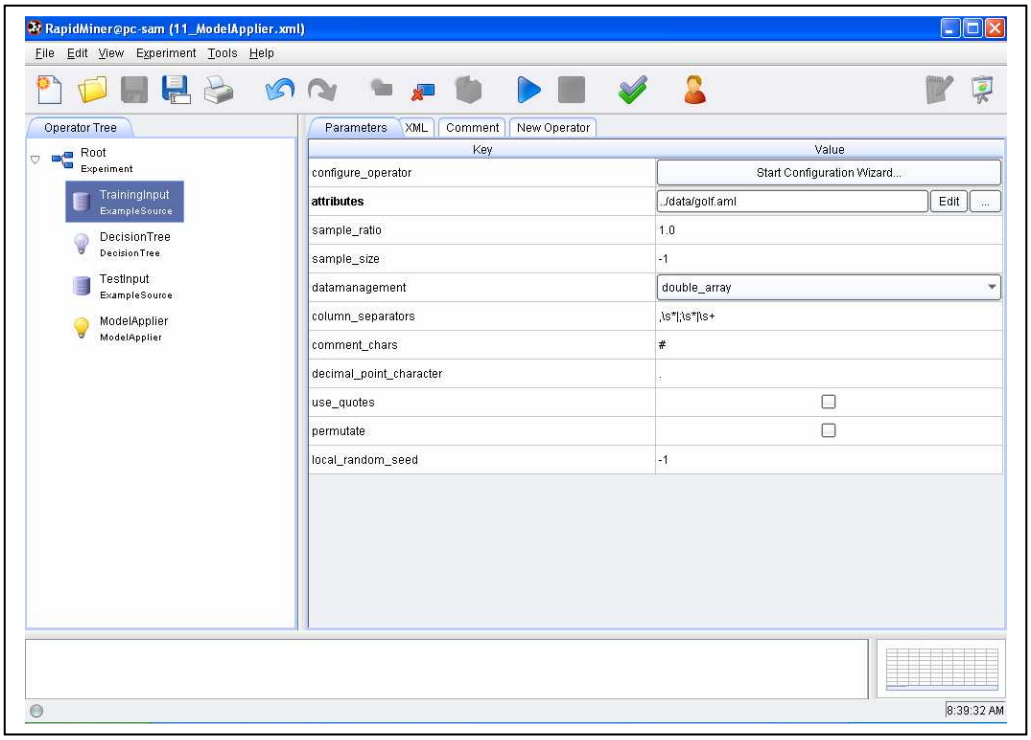


Figure 5.1 : Environnement RapidMiner

Figure 5.2 : Google Desktop

[1] Plusieurs moteurs de recherche locale existent sur le Net tels que Google Desktop, X1 de Yahoo, Wilber, etc.
[2] http://desktop.google.com/fr/

4. Déroulement des différentes phases de l'étude menée

4.1. Préparation des données

4.1.1. Evaluation des pertinences des documents par l'expertise humaine

Comme il a été mentionné à la section 2.1, chapitre 4, après la soumission de l'ensemble du corpus CCA entre les mains du groupe d'étudiants, ces derniers ont commencé par la lecture de l'ensemble des documents, puis ils ont rédigé les 21 requêtes qui ont été sujet à la recherche documentaire et enfin ils ont définit 334 documents pertinents pour l'ensemble de ces requêtes.

4.1.2. Processus de prétraitement et de préparation des textes

Dans ce qui suit, nous allons voir en détail tout le processus de prétraitement exposé dans la section 2.1 chapitre 4.

4.1.2.1. De l'XML vers le Texte

Dans une première étape, il fallait transformer le corpus dans une forme reconnaissable par l'outil de stemming que nous avons utilisé, à savoir *Al-stem* de *K. Darwish.* Etant donné que les textes de notre corpus sont rédigés sous XML avec un encodage UTF-8 *(Unicode Text Format-8)* (figure 5.3), nous avons utilisé *Microsoft Word édition 2002* pour les transformer en des textes au format textes (toujours sous l'encodage UTF-8).

4.1.2.2. Nettoyage de textes, Translittération et Tokenisation

Le nettoyage de textes a consisté à enlever toutes les séquences de caractères qui pouvaient engendrer un bruit dans le traitement (Balises XML, chiffres, caractères latins, symboles, …) et garder que les séquences de lettres arabes.

Dans une deuxième étape, nous avons effectué une translittération des séquences de lettres arabes à des séquences de lettres latines (format standard ASCII), en utilisant la translittération utilisée dans *AL-Stem* (tableau 5.3). Il existent d'autres translittérations, la plus connue est celle de *Tim Buckwalter* [Buckwalter, 2002][1].

Au cours de cette étape, les marques diacritiques sont éliminées et deux normalisations sont faites (normalisation agressive *aggressive normalization*[2]) :

ا, أ, إ, آ, ئ, ؤ, ء → A et ى, ي → y

[1] http://www.qamus.org/transliteration.htm

[2] Dans *AL-Stem*, deux normalisation sont possibles : la normalisation agressive *(aggressive normalization)* et la normalisation légère *(light normalization)* ا, أ, إ, آ → A

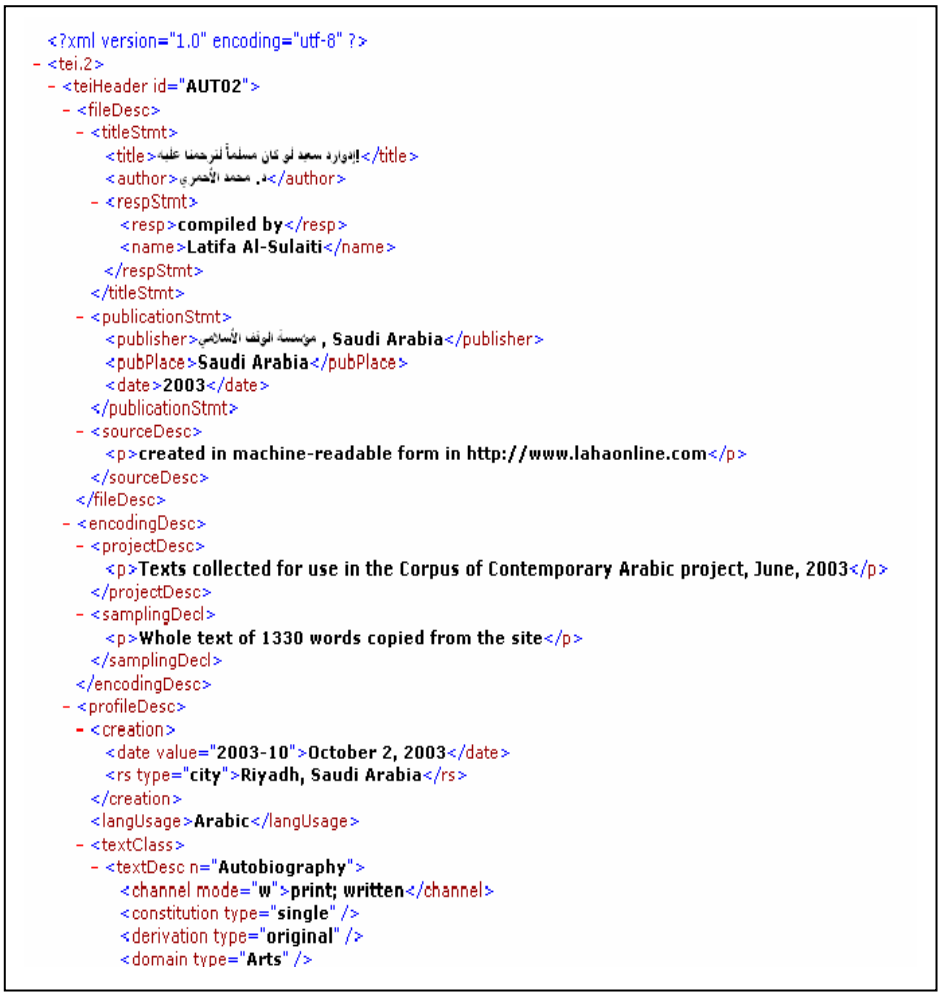


Figure 5.3 : Entête d'un fichier XML encodage UTF-8

Lettre arabe	Correspondant français		Lettre arabe	Correspondant français
ا, ئ, إ, ؤ, أ, آ, ء	A		ض	D
ب	b		ط	T
ت	t		ظ	Z
ة	p		ع	E
ث	v		غ	g
ج	j		ف	f
ح	H		ق	q
خ	x		ك	k
د	d		ل	l
ذ	O		م	m
ر	r		ن	n
ز	z		ه	h
س	s		و	w
ش	P		ي	y
ص	S		ى	y

Tableau 5.3 : Correspondance des lettres arabes-latines utilisée dans *Al-Stem*

Dans la troisième étape, une tokenisation est faite sur les séquences obtenues. L'élimination des *stopwords* se fait selon une liste contenue dans le fichier *Stoplist.txt* (annexe 2), fournie avec le stemmer *AL-Stem.* Ce fichier comprend 131 mots.

Les fichiers textes obtenus à la fin de cette phase (figure 5.4), sont appelés fichiers translittérés.

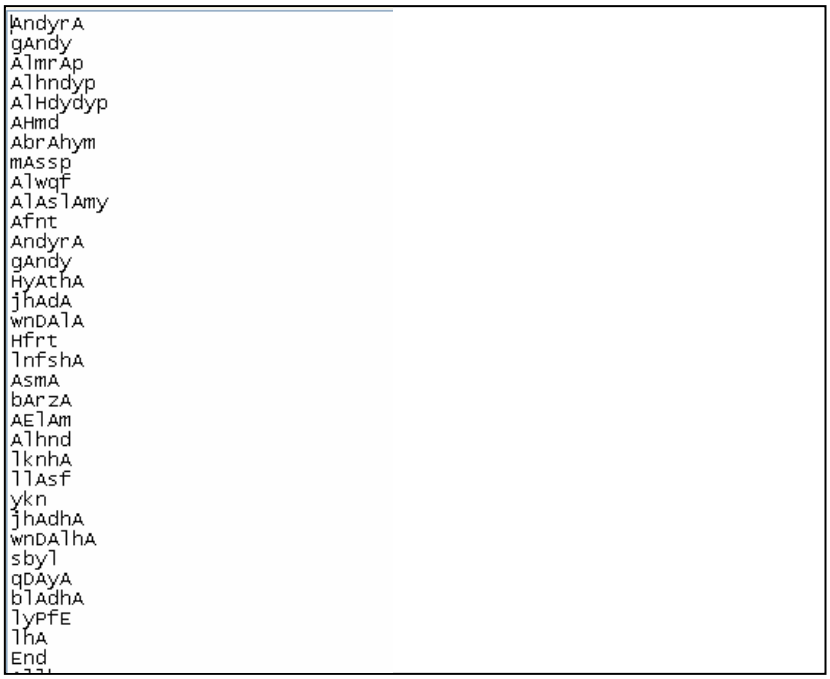


Figure 5.4 : Exemple de fichier translittéré

4.1.2.3. Stemming

Comme nous avons déjà vu dans la section 2.2, chapitre 1, le stemming consiste à rechercher un stem à mot en langue naturelle, en enlevant les préfixes et les suffixes (figure 5.5). Pour se faire, nous avons appliqué l'algorithme *Al-stem* (rédigé en langage *PERL*), qui utilise la liste des affixes fournies dans le tableau 3.6, sur les textes translittérés.

Notant que le choix du stemmer *Al-Stem* est dû, essentiellement, à sa performance en indexation [Darwish et al., 2005] qui avoisine les 60 % de réduction.

La figue 5.5 présente le fragment de code qui localise les affixes.

```
foreach $line (@lines) {
      if ($line =~ /^([wfb]Al|[bylmtwsn]t|[blwkf]m|[Al]l|[wlsf]y|[wflb]A|)(.*?)
            (At|wA|tA|wn|wh|An|ty|th|tm|km|h[nm]|hA|yp|tk|nA|y[nh]|[phyA]|)$/)
                  {
```

Figure 5.5 : Extrait du code de Stemming de *Al-Stem*

Explication :

Au début de chaque mot enlever (remplacement par un vide) :

Pour [wfb] Al:

le préfixe Al (ال)

les lettres w, f, b (و ف ب) si elles sont suivies par le préfixe Al (ال)

de même pour : [bylmtwsn]t,[blwkf]m,[Al]l, [wlsf]y,[wflb]A

A la fin de chaque mot enlever (remplacement par un vide) :

At,wA,tA,wn,wh,An,ty,th,tm,km,hA,tk,yp,nA,p,h,y,A

Pour h[nm]:

le suffixe h (ه)

les lettres n, m (ن م) si elles sont précédées par le suffixe h (ه)

de même pour y[nh]

Remarque importante

L'application de l'algorithme de stemming provoque par fois quelques aberrations telles que, par exemple, bArzA (بارزا) devient rz (رز) ce qui est absurde. Mais, heureusement, l'effet de ces aberrations est atténué par leurs relativités de présence dans les textes.

A la fin, Nous avons obtenu les fichiers que nous avons nommé fichiers stems (figure 5.6).

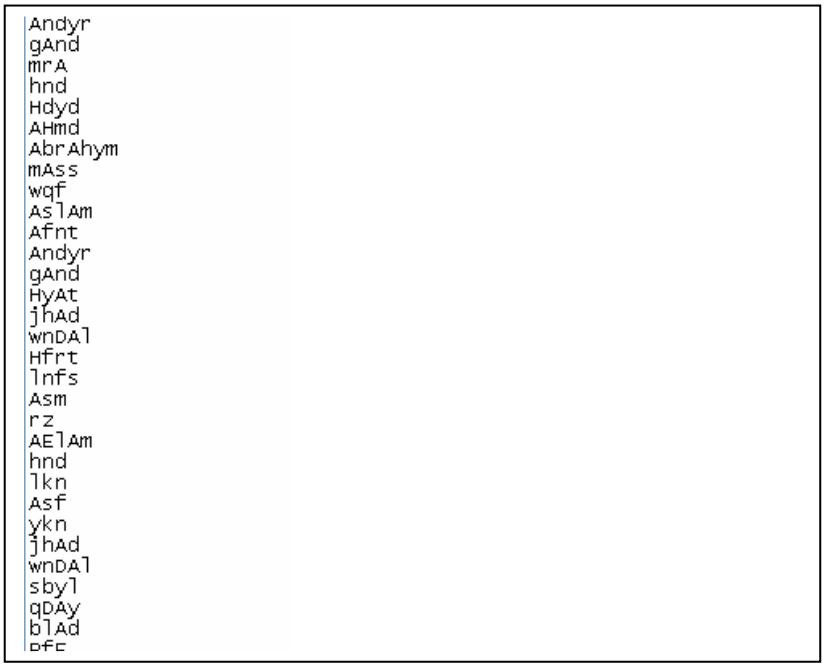


Figure 5.6 : Exemple de fichier stem

Il faut noter que nous avons apporté quelques modifications sur l'algorithme original *Al-Stem* afin de l'adapter avec nos besoins, ceci est, évidement, avec l'accord de son auteur. Les modifications apportées sont :

* Modification des arguments d'entrée et de sortie de l'algorithme original, afin qu'il puisse travailler avec des fichiers entiers à la place des mots. Ceci implique la modification de plusieurs paramètres et instructions dans l'algorithme original.

* Génération des fichiers intermédiaires : Fichiers Translittérés bruts, Fichiers Translittérés et Fichiers Stems.

* Utilisation de la normalisation agressive.

4.1.2.4. Indexation en vecteur TF-IDF

Au cours de cette étape, nous avons déroulé l'opérateur *WVTool* (figure 5.7) fourni avec le module *Text* de l'environnement YALE sur les deux ensembles des fichiers Translittérés et des fichiers Stems, pour obtenir en sortie les deux Datasets TF-IDF Translitté et TF-IDF Stem.

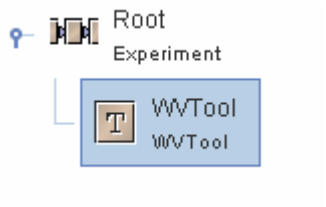
Expérimentation	Code XML
	<operator name="Root" class="Experiment"> <operator name="WVTool" class="WVTool"> <parameter key="id_attribute_type" value="short"/> <parameter key="min_chars" value="2"/> <list key="texts"> <parameter key="" value="F:\CorpusFinal\Translate"/> </list> </operator> </operator>

Figure 5.7 : Génération des vecteurs TF-IDF

Quatre paramètres de l'opérateur *WVTool* ont été configurés :

Source (*texts=Chemin de la collection*)

Codage (*vectorcreation = TFIDF*),

Longueur de mots pris en compte (*min_chars=2*)

Prise en compte des noms des fichiers (*id_attribut_type=short*)

Résultats de l'indexation

La génération des Datasets TF-IDF Translitté et TF-IDF Stem de taille respectivement, 92.732 attributs et 34.584 attributs, nous a permis de constater une réduction

de 62 % du nombre d'attributs, ce qui confirme que le stemming est une autre forme de réduction de dimension.

4.2. Clustering documentaire

4.2.1. Génération des différentes partitions

Pour générer les clusters des deux collections (Translittéré et Stem), nous avons utilisé l'opérateur *AgglomérativeFlatClustering* (figure 5.8) pour implémenter la méthode hiérarchique ascendante et l'opérateur *KMedoids* pour implémenter la méthode PAM (figure 5.9).

Plusieurs autres opérateurs ont été aussi utilisés dans les expérimentations, nous citons:

Exemplesource: Utilisé pour la sélection de la source de données (dans notre cas le datasets TF-IDF Translittéré ou datasets TF-IDF Stem.

ExempleSet2Similarity et *ClusterDensityEvaluator*: Ces deux opérateurs sont utilisés ensemble pour l'évaluation de la densité des clusters générés.

ParameterIteration: Joue le rôle d'une boucle. Il est utilisé pour faire varié plusieurs paramètres, le plus important, dans ce cas, est le nombre K de clusters générés. Il est configuré à 5, 7, 9, 11, 13, 15 clusters.

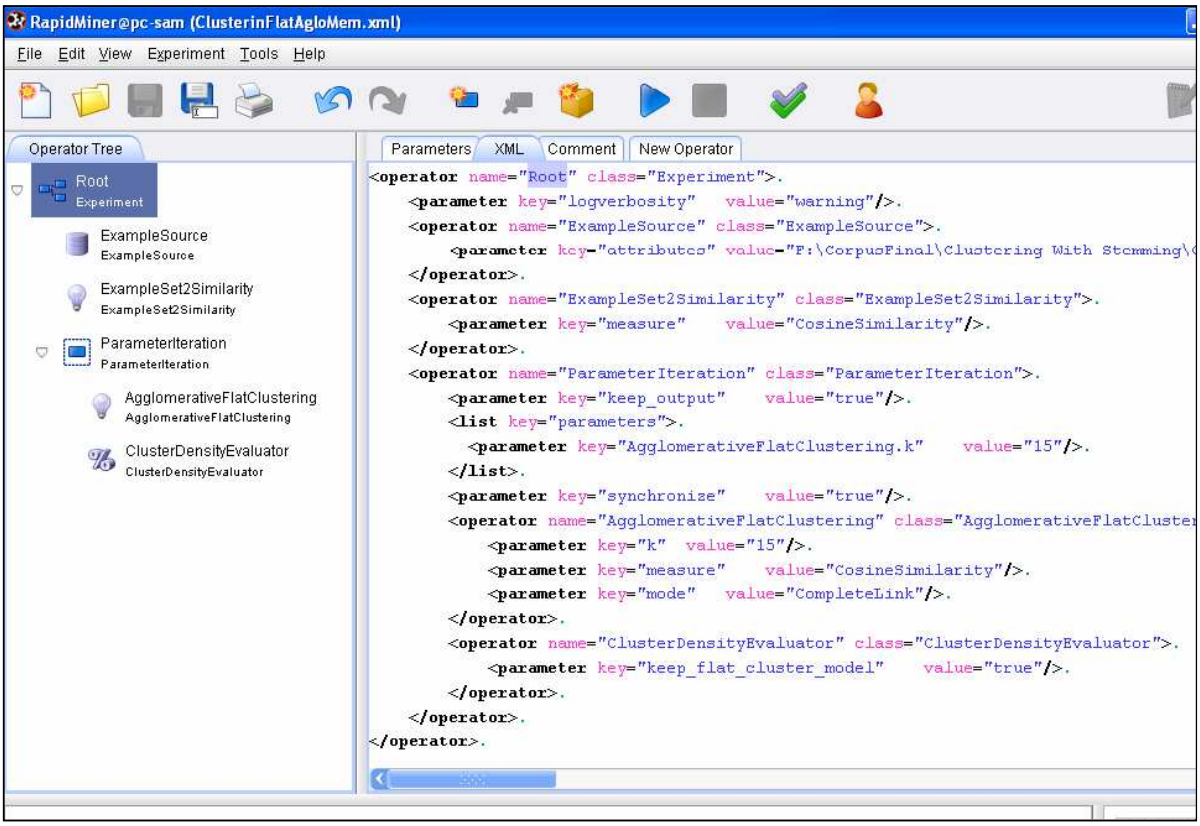


Figure 5.8 : Implémentation de la méthode agglomérative ascendante

Deux paramètres de l'opérateur *AgglomérativeFlatClustering* ont été configurés :

Mesure de similarité (*Mesure=CosinusSimilarity*)

Mode (*Mode = CompleteLink*),

Pour l'opérateur *KMedoids*, trois paramètres ont été configurés :

Mesure de similarité (*Mesure=CosinusSimilarity*)

Nombre maximum de changement d'un médoïd par un non-médoïd (*Max_Run = 5*)

Nombre maximum d'itérations avec K médoïds actuel (*Max_Optimization_Step=100*)

Pour les deux derniers paramètres, plusieurs valeurs ont été testées. La fixation aux valeurs, respectivement, 5 et 100 est faite après la stabilisation des clusters, c'est-à-dire qu'aucun déplacement de documents entre clusters n'a été détecté.

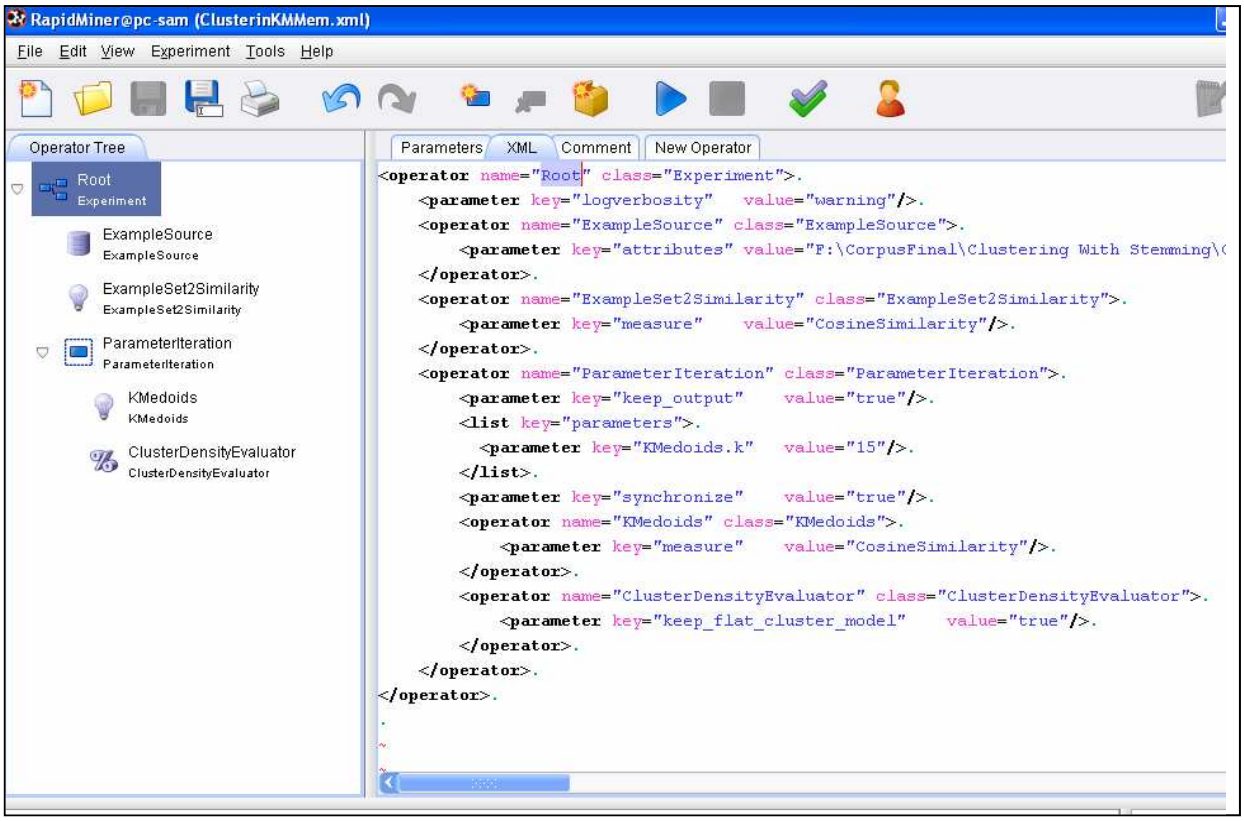


Figure 5.9 : Implémentation de la méthode des K-médoïds

4.2.2. Qualité des clusters générés

Pour évaluer la qualité des clusters générés, nous avons utilisé la mesure de la similarité globale décrite dans la section 3.1.4, chapitre 2 (en utilisant les deux opérateurs *ExempleSet2Similarity et ClusterDensityEvaluator*). Les résultats obtenus, sont synthétisés dans le tableau 5.4.

Nbr de clusters	5	7	9	11	13	15
AggloAsc Avec stemming	0,081	0,088	0,093	0,123	0,145	0,152
AggloAsc Sans stemming	0,046	0,068	0,074	0,087	0,091	0,114
K-Med Avec stemming	0,083	0,085	0,099	0,112	0,142	0,140
K-Med Sans stemming	0,054	0,059	0,084	0,098	0,101	0,108

Tableau 5.4 : Moyenne de la similarité globale *(overall similarity)* des clusters

Du tableau 5.4 et de la figure 5.10, nous remarquons que le stemming améliore la similarité globale *(overall similarity)* dans les clusters obtenus (amélioration d'environ 3 % dans les deux méthodes utilisées), ceci est dû essentiellement à l'effet du stemming qui a aidé à enlever la flexion des mots qui ont la même racine, donc les documents qui se rapportent à une même thématique auront plus de chance de se retrouver dans le même cluster.

Nous remarquons aussi qu'avec l'augmentation du nombre de clusters, la similarité globale augmente et les clusters deviennent de plus en plus denses, ce qui nous renseigne sur la diversité du corpus. Une troisième observation est que nul des deux méthodes de clustering utilisées ne s'est nettement imposée.

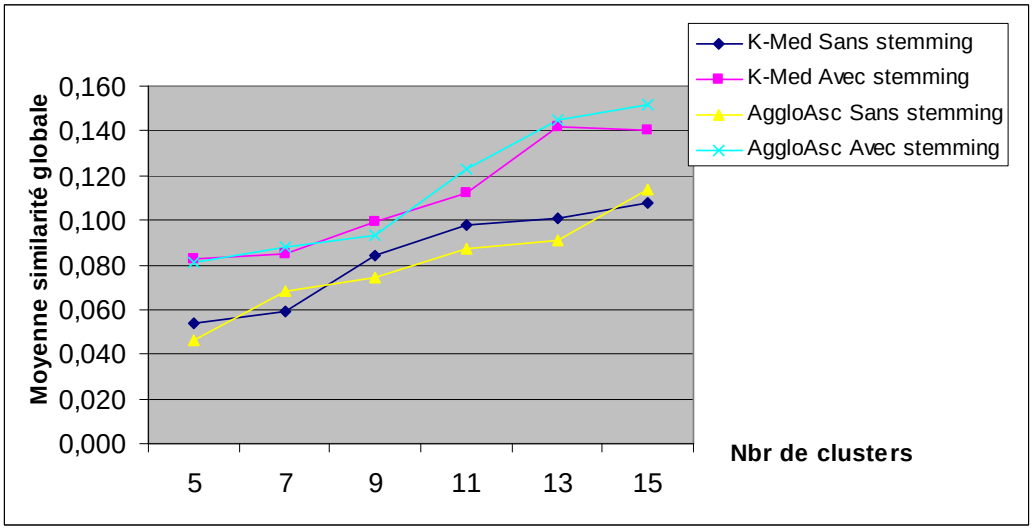
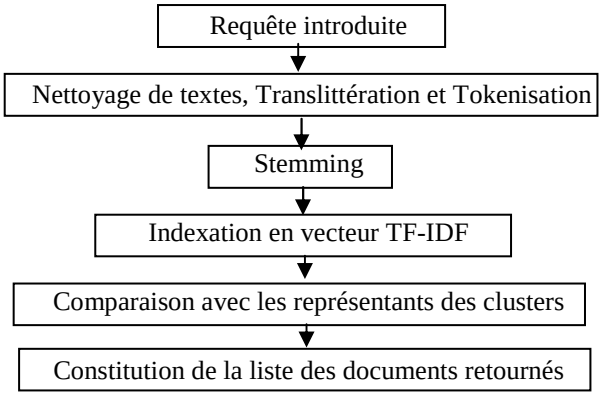


Figure 5.10 : Moyenne de la similarité globale *(overall similarity)* des clusters

4.3. Application de l'approche proposée

L'application de l'approche proposée s'est déroulée selon l'organigramme suivant :



4.3.1. Traitement d'une requête

La requête étant traitée comme un document, elle doit subir donc un traitement de nettoyage, de translittération, de tokenisation, de stemming et de représentation en vecteur TF-IDF. Par exemple, si nous prenons la requête numéro 3 à l'état brut :

ما هي العلاقات التي قد توجد بين السياحة و عالم الطيران و السفر الجوي

devient après le processus de prétraitement:

Forme translittérée:

```
AlElAqAt Alty twjd AlsyAHp EAlm AlTyrAn Alsfr Aljwy
```

Forme Stem:

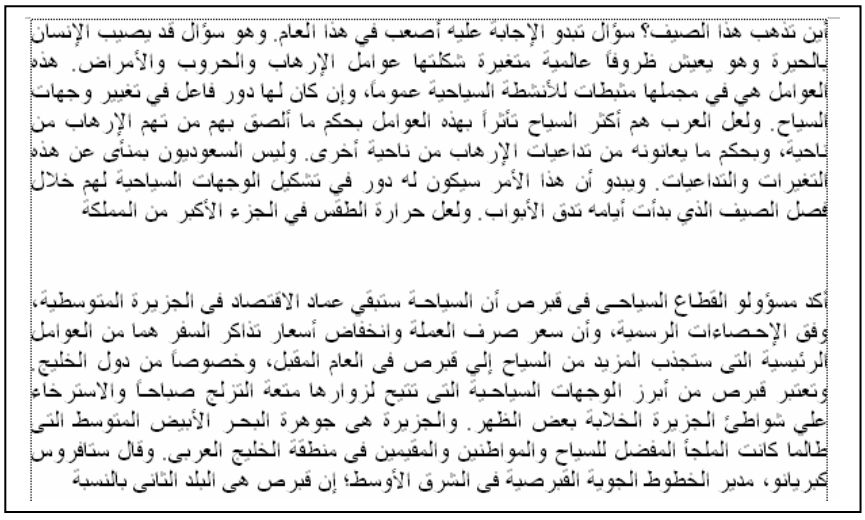
```
ElAq twjd syAH EAlm Tyr sfr jw
```

4.3.2. Constitution de la liste des documents retournés

Pour constituer la liste de documents retournés, nous avons appliqué l'algorithme 4.1 pour chaque requête introduite.

Par exemple, pour la requête 3 décrite dans l'exemple précédent et une partition en 13 clusters avec la méthode par agglomération ascendante, les représentants des 13 clusters sont : *To31_Stem.txt, Spo02_Stem.txt, Sc43_Stem.txt, HM29_Stem.txt, To22_Stem.txt, HM10_Stem.txt, Rel10_Stem.txt, Ec06_Stem.txt, Sc34_Stem.txt, Rec09_Stem.txt, CHD20_Stem.txt, CHD20_Stem.txt* et *AUT45_Stem.txt.*

Le représentant le plus proche (par calcul de distance) de la requête 3 est *To31_Stem.txt* (figure 5.11), donc la liste des documents retournée (annexe 3) va commencer avec ce document et va comprendre les fichiers qui composent le cluster qui a comme représentant ce document.



أين تذهب هذا الصيف؟ سؤال تبدو الإجابة عليه أصعب في هذا العام. وهو سؤال قد يصيب الإنسان بالحيرة وهو يعيش ظروفاً عالمية متغيرة شكلتها عوامل الإرهاب والحروب والأمراض. هذه العوامل هي في مجملها مثبطات للأنشطة السياحية عموماً، وإن كان لها دور فاعل في تغيير وجهات السياح. ولعل العرب هم أكثر السياح تأثراً بهذه العوامل بحكم ما ألصق بهم من تهم الإرهاب من ناحية، وبحكم ما يعانونه من تداعيات الإرهاب من ناحية أخرى. وليس السعوديون بمنأى عن هذه المتغيرات والتداعيات. ويبدو أن هذا الأمر سيكون له دور في تشكيل الوجهات السياحية لهم خلال فصل الصيف الذي بدأت أيامه تدق الأبواب. ولعل حرارة الطقس في الجزء الأكبر من المملكة

أكد مسؤولو القطاع السياحي في قبرص أن السياحة ستبقي عماد الاقتصاد في الجزيرة المتوسطية، وفق الإحصاءات الرسمية، وأن سعر صرف العملة وانخفاض أسعار تذاكر السفر هما من العوامل الرئيسية التي ستجذب المزيد من السياح إلي قبرص في العام المقبل، وخصوصاً من دول الخليج وتعتبر قبرص من أبرز الوجهات السياحية التي تتيح لزوارها متعة التزلج صباحاً والاسترخاء على شواطئ الجزيرة الخلابة بعض الظهر. والجزيرة هي جوهرة البحر الأبيض المتوسط التي طالما كانت الملجأ المفضل للسياح والمواطنين والمقيمين في منطقة الخليج العربي. وقال ستافروس كيريانو، مدير الخطوط الجوية القبرصية في الشرق الأوسط؛ إن قبرص هي البلد الثاني بالنسبة

Figure 5.11 : Extraits des fichiers *To31_Stem.txt* et *To53_Stem.txt* par lesquels commence la liste retournée en réponse à la requête 3

Ensuite, cette liste est complétée par les fichiers des autres clusters et selon l'algorithme 4.1. Dans notre cas le représentant le plus proche de *To31_Stem.txt* est *To22_Stem.txt* puis le représentant *Ec06_Stem.txt* et ainsi de suite.

4.3.3. Evaluation de la liste des documents retournée

Les deux mesures d'évaluation (la précision et le rappel) sont calculées par rapport à la liste des documents pertinents pour chaque requête (annexe 1).

Par exemple pour la requête 3 (annexe 3), nous avons obtenu une précision de 0.60 et un rappel de 0.06 pour les cinq premiers documents retournés. Pour les dix premiers documents retournés, nous avons obtenu une précision de 0.70 et un rappel de 0.20, etc.

Les deux tableaux 5.5 et 5.6 représentent la précision moyenne et le rappel moyen des listes de documents renvoyés (les 5, 10, 20, 30, 40, 50, 100, et 200 premiers documents) en réponse aux 21 requêtes et ceci avec les deux méthodes de clustering choisies. Dans le premier tableau, l'étude concerne les partitions en 5, 7 et 9 clusters, dans le deuxième, elle concerne les partitions en 11, 13 et 15 clusters.

Nbr documents retournés	**5**			**10**			**20**			**30**			**40**			**50**			**100**			**200**		
Nbr Clusters	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**
Avec Stemming																								
AggloAsc																								
Rappel	0,07	0,07	0,11	0,14	0,12	0,16	0,22	0,18	0,24	0,30	0,21	0,32	0,37	0,29	0,43	0,49	0,38	0,52	0,84	0,94	0,81	0,92	0,94	1,00
Précision	0,21	0,19	0,25	0,22	0,18	0,21	0,19	0,15	0,18	0,18	0,13	0,17	0,16	0,13	0,18	0,16	0,13	0,17	0,14	0,16	0,14	0,08	0,08	0,08
K-Med																								
Rappel	0,11	0,12	0,08	0,17	0,21	0,13	0,27	0,29	0,25	0,38	0,36	0,38	0,46	0,41	0,51	0,52	0,52	0,59	0,75	0,76	0,78	0,92	0,94	0,93
Précision	0,30	0,24	0,23	0,25	0,23	0,21	0,22	0,20	0,20	0,21	0,17	0,21	0,19	0,16	0,21	0,18	0,17	0,19	0,12	0,12	0,12	0,08	0,08	0,07
Sans Stemming																								
AggloAsc																								
Rappel	0,03	0,06	0,07	0,07	0,11	0,13	0,14	0,18	0,23	0,17	0,24	0,34	0,23	0,29	0,43	0,31	0,37	0,51	0,64	0,65	0,73	0,83	1,00	0,99
Précision	0,11	0,15	0,21	0,13	0,16	0,21	0,13	0,16	0,20	0,11	0,16	0,21	0,11	0,14	0,19	0,12	0,14	0,18	0,12	0,12	0,13	0,07	0,08	0,08
K-Med																								
Rappel	0,04	0,06	0,10	0,09	0,10	0,20	0,15	0,16	0,31	0,20	0,24	0,47	0,23	0,31	0,53	0,30	0,38	0,62	0,54	0,62	0,94	0,98	0,92	1,00
Précision	0,14	0,15	0,29	0,14	0,14	0,27	0,13	0,12	0,22	0,12	0,13	0,23	0,11	0,13	0,20	0,12	0,12	0,19	0,09	0,10	0,15	0,08	0,07	0,08

Tableau 5.5 : Précision et Rappel moyens de l'approche proposée avec 5, 7 et 9 clusters

Nbr documents retournés	**5**			**10**			**20**			**30**			**40**			**50**			**100**			**200**		
Nbr Clusters	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**
Avec Stemming																								
AggloAsc																								
Rappel	0,14	0,15	0,16	0,27	0,32	0,31	0,39	0,43	0,42	0,57	0,63	0,62	0,65	0,72	0,70	0,70	0,80	0,77	0,85	0,90	0,88	0,98	0,98	0,98
Précision	0,43	0,49	0,49	0,44	0,50	0,48	0,34	0,36	0,34	0,34	0,36	0,34	0,29	0,31	0,30	0,25	0,27	0,27	0,15	0,15	0,15	0,08	0,08	0,08
K-Med																								
Rappel	0,14	0,13	0,12	0,27	0,26	0,25	0,44	0,45	0,38	0,60	0,62	0,57	0,70	0,68	0,67	0,77	0,74	0,76	0,92	0,84	0,85	1,00	1,00	1,00
Précision	0,42	0,42	0,35	0,40	0,41	0,36	0,36	0,38	0,31	0,33	0,34	0,32	0,29	0,28	0,28	0,26	0,25	0,26	0,15	0,14	0,15	0,08	0,08	0,08
Sans Stemming																								
AggloAsc																								
Rappel	0,16	0,18	0,13	0,25	0,27	0,25	0,43	0,47	0,43	0,57	0,64	0,60	0,64	0,70	0,67	0,71	0,78	0,75	0,81	0,96	0,96	0,88	1,00	1,00
Précision	0,40	0,46	0,40	0,34	0,37	0,38	0,32	0,35	0,34	0,30	0,33	0,33	0,26	0,28	0,28	0,24	0,26	0,26	0,14	0,15	0,16	0,07	0,08	0,08
K-Med																								
Rappel	0,14	0,11	0,11	0,25	0,21	0,23	0,47	0,41	0,37	0,64	0,54	0,57	0,75	0,69	0,67	0,82	0,78	0,76	0,92	0,90	0,90	0,93	0,91	0,92
Précision	0,36	0,33	0,35	0,34	0,33	0,35	0,35	0,35	0,31	0,33	0,31	0,32	0,29	0,29	0,28	0,26	0,26	0,26	0,15	0,18	0,17	0,09	0,09	0,09

Tableau 5.6 : Précision et Rappel moyens de l'approche proposée avec 11, 13 et 15 clusters

4.3.4. Influence du nombre de clusters sur une recherche documentaire avec l'approche proposée

Pour les deux méthodes de clustering utilisées, la difficulté majeure réside dans le choix du nombre de clusters qui doit être effectué à priori. Nous avons alors testé notre approche sur plusieurs partitions. Pour un nombre de clusters compris entre 5 et 15 (5,7,9,11,13 et 15) , la moins bonne précision moyenne est obtenue avec 7 clusters (0,18) et la meilleure avec 13 clusters (0,50) (tableau 5.7, figure 5.12) et ceci en utilisant un clustering avec agglomération ascendante sur des textes stemmés. Pour un clustering avec la méthode des K-médoïds, la moins bonne précision moyenne est obtenue avec 9 clusters (0,21) et la meilleure avec 13 clusters (0,41).

De la figure 5.12, nous remarquons qu'en général, l'amélioration de la précision avec l'augmentation du nombre de clusters. Mais nous remarquons aussi que la meilleur précision est obtenu avec la partition en 13 clusters, ce qui veut dire que le meilleur résultat ne correspond pas toujours au nombre élevé de clusters.

Avec des textes en brut, la moins bonne précision moyenne est obtenue avec 5 clusters (0,13) et la meilleure avec 15 clusters (0,38) (tableau 5.7, figure 5.12) en utilisant la première méthode de clustering. Avec la deuxième méthode, la moins bonne précision moyenne est avec 5 et 7 clusters (0,14) et la meilleure est avec 15 clusters (0,35). Ceci confirme le résultat obtenu avec les textes stemmés, c'est-à-dire, l'amélioration de la précision avec l'augmentation du nombre de clusters.

Enfin, dans les deux cas (textes stemmés et textes bruts), l'influence du processus de clustering sur la qualité d'une recherche documentaire est flagrante, ceci est dû essentiellement, et à notre avis, à la qualité des clusters obtenus avec les deux techniques de clustering.

4.3.5. Influence du stemming sur une recherche documentaire avec l'approche proposée

Quel que soit la méthode utilisée pour le clustering, en faisant une comparaison entre les résultats obtenus sur des textes stemmés et des textes à l'état brut, nous avons remarqué que le stemming agit directement et positivement sur la qualité de la recherche, soit une amélioration dans la précision moyenne de 7.3 % pour la première méthode et de 4.8 % pour la deuxième (tableau 5.7, figure 5.12), ceci est directement liée à la qualité des clusters générés.

Méthode de clustering	Nbr de clusters					
	5	7	9	11	13	15
AggloAsc avec stemming	0,22	0,18	0,21	0,44	0,50	0,48
K-Med avec stemming	0,25	0,23	0,21	0,40	0,41	0,36
AggloAsc sans stemming	0,13	0,16	0,21	0,34	0,37	0,38
K-Med sans stemming	0,14	0,14	0,27	0,34	0,33	0,35

Tableau 5.7 : Précision moyenne des 10 premiers documents retournés avec l'approche proposée sur plusieurs clusters

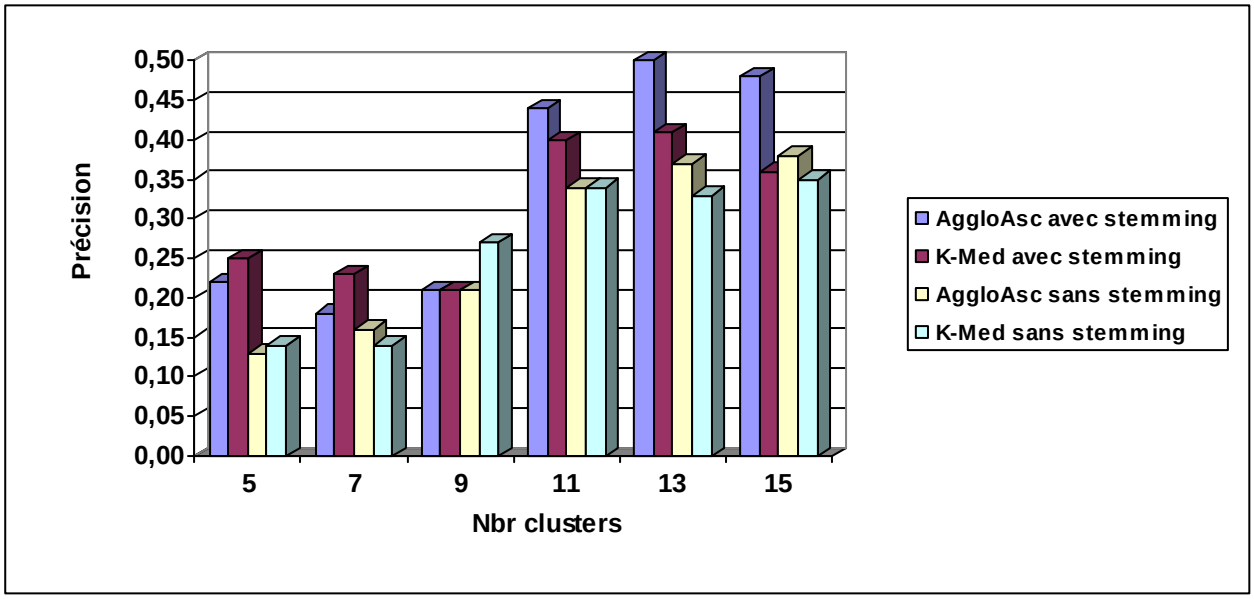


Figure 5.12 : Influence du nombre de clusters sur une recherche avec l'approche proposée

4.4. Expérimentation en utilisant un moteur classique de recherche locale

Ici, la liste des documents renvoyée est celle fournie par le moteur classique *Google desktop* après l'introduction de la requête sous la forme translittérée ou sous la forme Stem.

La figure 5.13 montre la liste des documents renvoyée par le moteur de recherche en réponse à la requête 3 (exemple de la section 4.3.1).

Remarque

Afin d'améliorer le résultat du moteur Google desktop, nous avons introduit la requête en plus de plusieurs combinaisons de mots constituant cette dernière en enlevant chaque fois un des mots. Nous avons essayé par ce procédé, d'imiter le fonctionnement des moteurs de recherche sur le Net.

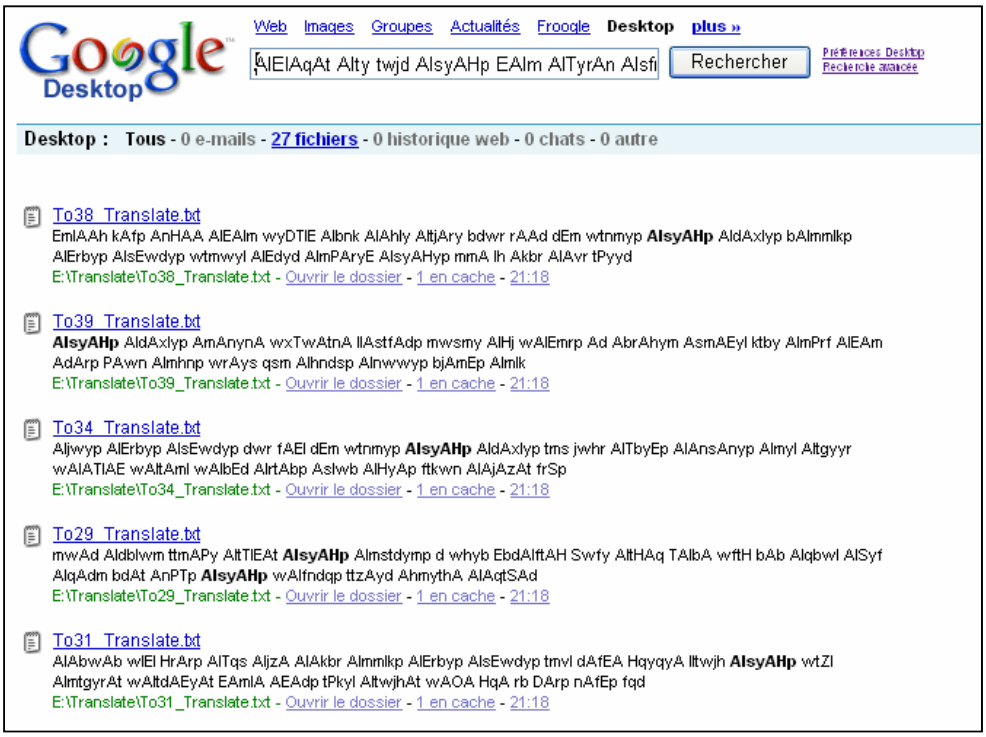


Figure 5.13 : Réponse du moteur de recherche classique à une requête

4.4.1. Influence du stemming sur une recherche classique

En lançant une recherche classique sur un corpus ayant subi un traitement de stemming, nous avons constaté une amélioration en précision moyenne de 3 % (tableau 5.8) et en rappel moyen de 4 % (tableau 5.9) par rapport à une recherche effectuée sur le même corpus mais à l'état brut, et cela, sur les 50 premiers documents renvoyés (figure 5.14).

La recherche classique étant basée essentiellement sur le *ranking*, compare les termes de la requête introduite avec ceux qui composent la représentation du document, elle ne rend que les documents contenant exactement les termes de la requête, excluant ainsi, tous les autres documents qui peuvent être pertinents pour cette requête et qui contiennent les flexions possibles des termes de la requête. Avec le stemming, la recherche classique place plus de documents jugés pertinents en avant de la liste retournée, cette performance est due essentiellement, selon notre avis, à l'atténuation du caractère flexionnel de la langue arabe par le stemming, qui ramène une grande variété de flexions des mots en un nombre réduit de stems.

Remarque

*Ici, Il est très important de noter, qu'à notre avis, le grand nombre de stems «**corrects**» qui peuvent être tirés du processus de stemming utilisé compense la perte en sémantique (section 3.1 chapitre 3).*

Précision moyenne

	à 5 documents retournés	à 10 premiers documents retournés	A 20 premiers documents retournés	à 30 premiers documents retournés	à 40 premiers documents retournés	à 50 premiers documents retournés
Avec Stemming	0,43	0,34	0,26	0,21	0,20	0,19
Sans Stemming	0,39	0,31	0,24	0,20	0,17	0,15

Tableau 5.8 : Précision moyenne des 50 premiers documents retournés par une recherche classique avec et sans stemming

Rappel moyen

	A 5 documents retournés	à 10 premiers documents retournés	à 20 premiers documents retournés	à 30 premiers documents retournés	à 40 premiers documents retournés	à 50 premiers documents retournés
Avec Stemming	0,17	0,25	0,37	0,43	0,51	0,60
Sans Stemming	0,15	0,23	0,33	0,40	0,45	0,49

Tableau 5.9 : Rappel moyen des 50 premiers documents retournés par une recherche classique avec et sans stemming

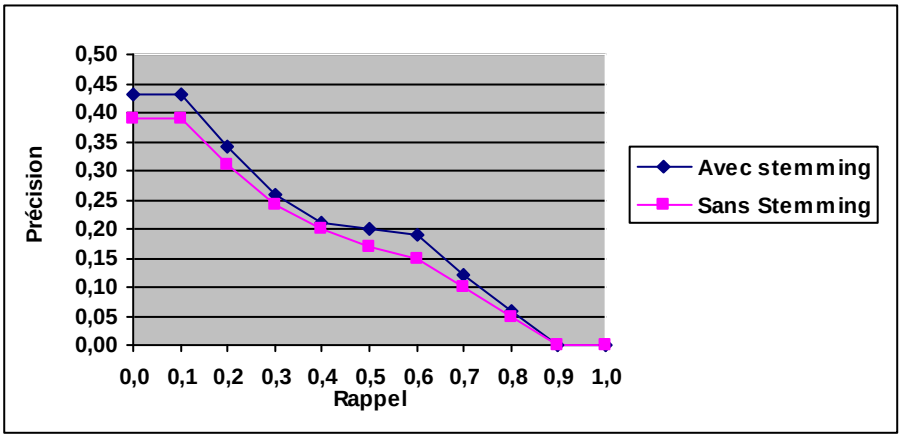


Figure 5.14 : Influence du stemming sur une recherche classique

4.5. Amélioration apportée avec l'approche proposée sur une recherche documentaire

A partir des résultats obtenus avec les différentes partitions, nous avons effectué une comparaison entre une moyenne de la précision et du rappel obtenus avec les 3 meilleurs partitions (moyenne de 11, 13 et 15 clusters) des 10 premiers documents rapportés par l'approche proposée (en utilisant les deux méthodes de clustering) et ceux obtenus par le système de recherche classique (tableaux 5.10 et tableau 5.11), puis une comparaison entre les résultats obtenus par l'approche proposée avec une partition en 13 clusters et ceux obtenus avec une recherche classique (tableaux 5.12 et tableau 5.13).

Le choix dans la deuxième évaluation de la partition en 13 clusters pour la comparaison est dû, essentiellement, à la performance des deux méthodes de clustering avec cette partition (corpus ayant subis un stemming).

4.5.1. Avec une moyenne des 11, 13 et 15 clusters

Dans cette comparaison, nous avons constaté une amélioration en précision estimée à 13 % pour la première méthode (tableau 5.10, figure 5.15) et à 5 % pour la deuxième et ceci par rapport au système de recherche classique. D'autre part, une amélioration en rappel estimée à 5 % pour la première méthode (tableau 5.11, figure 17) et à 1 % pour la deuxième.

Ces résultas étant pour un corpus ayant subis un stemming, les résultats pour le même corpus à l'état brut représentent une amélioration en précision estimée à 5 % pour la première méthode (tableau 5.10, figure 5.16) et à 3 % pour la deuxième, ceci toujours par rapport au système de recherche classique. D'autre part, une amélioration en rappel estimée à 3 % pour la première méthode (tableau 5.11, figure 18). La deuxième méthode n'a apporté aucune amélioration.

Précision moyenne

	à 5 documents retournés	à 10 documents retournés	à 20 documents retournés	à 30 documents retournés	à 40 documents retournés	à 50 documents retournés	à 100 documents retournés	à 200 documents retournés
AggloAsc avec stemming	0,47	0,47	0,35	0,35	0,30	0,26	0,15	0,08
K-Med avec stemming	0,40	0,39	0,35	0,33	0,29	0,26	0,15	0,08
Google Desktop	0,43	0,34	0,26	0,21	0,20	0,19	0,12	0,06
AggloAsc sans stemming	0,42	0,36	0,34	0,32	0,27	0,25	0,15	0,08
K-Med sans stemming	0,35	0,34	0,33	0,32	0,29	0,26	0,17	0,09
Google Desktop	0,39	0,31	0,24	0,20	0,17	0,15	0,10	0,05

Tableau 5.10 : Précision moyenne (des 11, 13 et 15 clusters) des premiers documents retournés

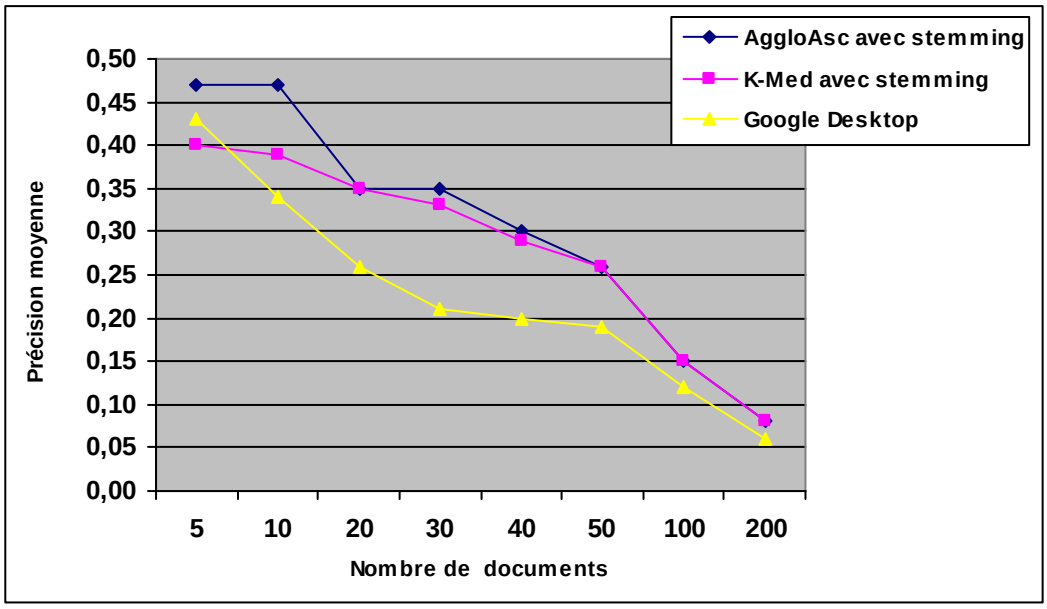


Figure 5.15 : Précision moyenne (des 11, 13 et 15 clusters) des premiers documents retournés (avec stemming)

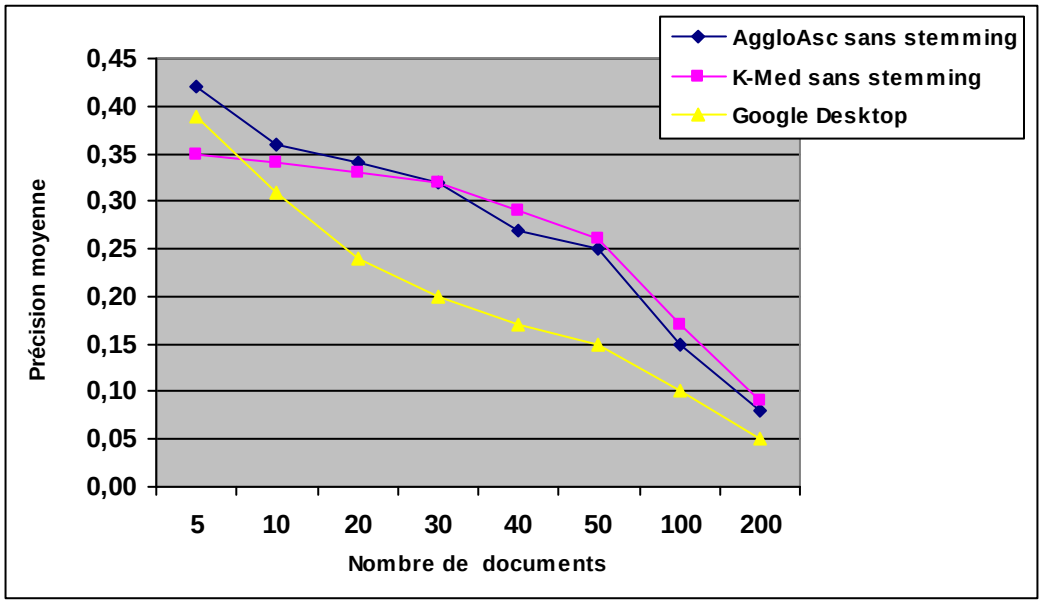


Figure 5.16 : Précision moyenne (des 11, 13 et 15 clusters) des premiers documents retournés (sans stemming)

Rappel moyen

	à 5 documents retournés	à 10 documents retournés	à 20 documents retournés	à 30 documents retournés	à 40 documents retournés	à 50 documents retournés	à 100 documents retournés	à 200 documents retournés
AggloAsc avec stemming	0,15	0,30	0,41	0,61	0,69	0,76	0,88	0,98
K-Med avec stemming	0,13	0,26	0,42	0,60	0,68	0,75	0,87	1,00
Google Desktop	0,17	0,25	0,37	0,43	0,51	0,60	0,75	0,82
AggloAsc sans stemming	0,16	0,26	0,45	0,60	0,67	0,75	0,91	0,96
K-Med sans stemming	0,12	0,23	0,42	0,58	0,70	0,79	0,91	0,92
Google Desktop	0,15	0,23	0,33	0,40	0,45	0,49	0,58	0,63

Tableau 5.11 : Rappel moyen (des 11, 13 et 15 clusters) des premiers documents retournés

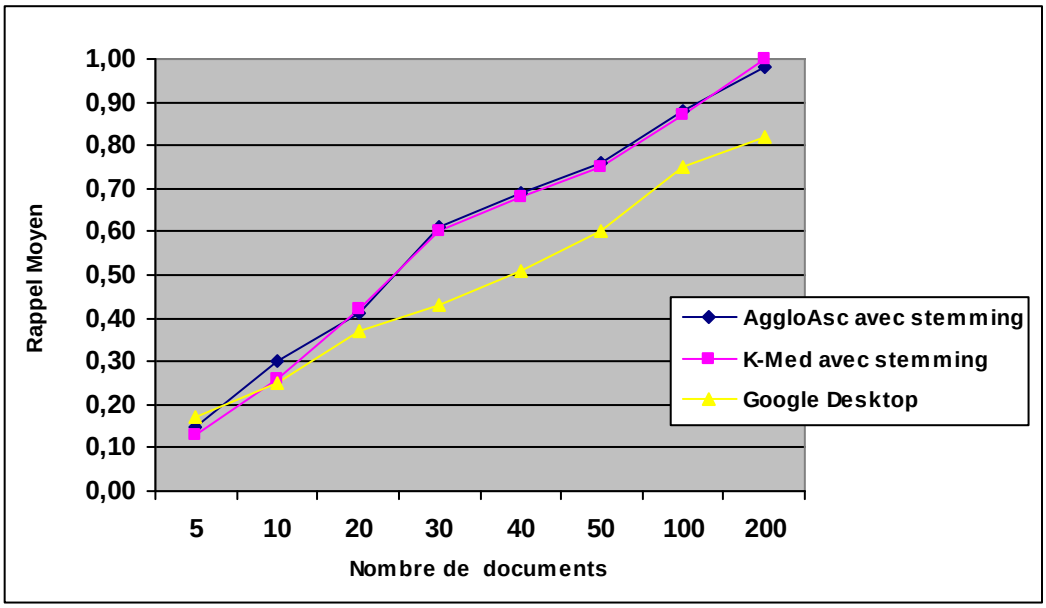


Figure 5.17 : Rappel moyen (des 11, 13 et 15 clusters) des premiers documents retournés (avec stemming)

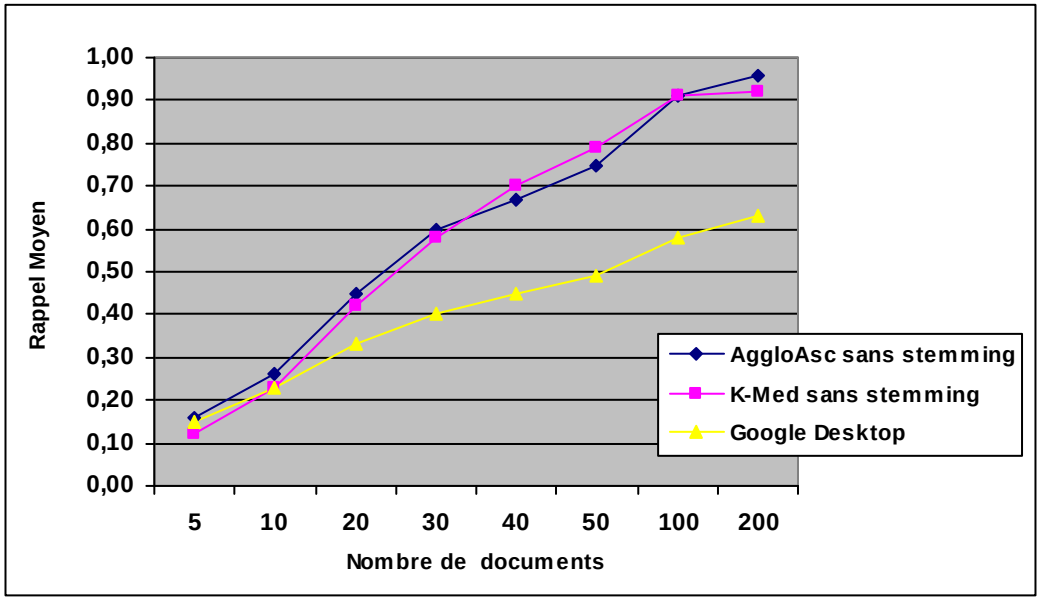


Figure 5.18 : Rappel moyen (des 11, 13 et 15 clusters) des premiers documents retournés (sans stemming)

4.5.2. Avec 13 clusters

Dans cette comparaison nous avons constaté, également, une amélioration en précision estimée à 16 % pour la première méthode de clustering (tableau 5.12, figure 5.19) et à 7 % pour la deuxième et ceci toujours par rapport au système de recherche classique. D'autre part, une amélioration en rappel estimée à 7 % pour la première méthode (tableau 5.13, figure 5.20) et à 1 % pour la deuxième.

Ces résultas étant pour un corpus ayant subis un stemming, les résultats pour le même corpus à l'état brut représentent une amélioration en précision estimée à 6 % pour la première méthode (tableau 5.12, figure 5.21) et à 2 % pour la deuxième et ceci par rapport au système de recherche classique. D'autre part, une amélioration en rappel estimée à 4 % pour la première méthode (tableau 5.13, figure 5.22) et une dégradation estimée à 2 % par rapport au système de recherche classique.

Précision moyenne

	à 5 documents retournés	à 10 documents retournés	à 20 documents retournés	à 30 documents retournés	à 40 documents retournés	à 50 documents retournés	à 100 documents retournés	à 200 documents retournés
AggloAsc avec stemming	0,49	0,50	0,36	0,36	0,31	0,27	0,15	0,08
K-Med avec stemming	0,42	0,41	0,38	0,34	0,28	0,25	0,14	0,08
Google Desktop	0,43	0,34	0,26	0,21	0,20	0,19	0,12	0,06
AggloAsc sans stemming	0,46	0,37	0,35	0,33	0,28	0,26	0,15	0,08
K-Med sans stemming	0,33	0,33	0,35	0,31	0,29	0,26	0,18	0,09
Google Desktop	0,39	0,31	0,24	0,20	0,17	0,15	0,10	0,05

Tableau 5.12 : Précision moyenne (partition en 13 clusters) des premiers documents retournés

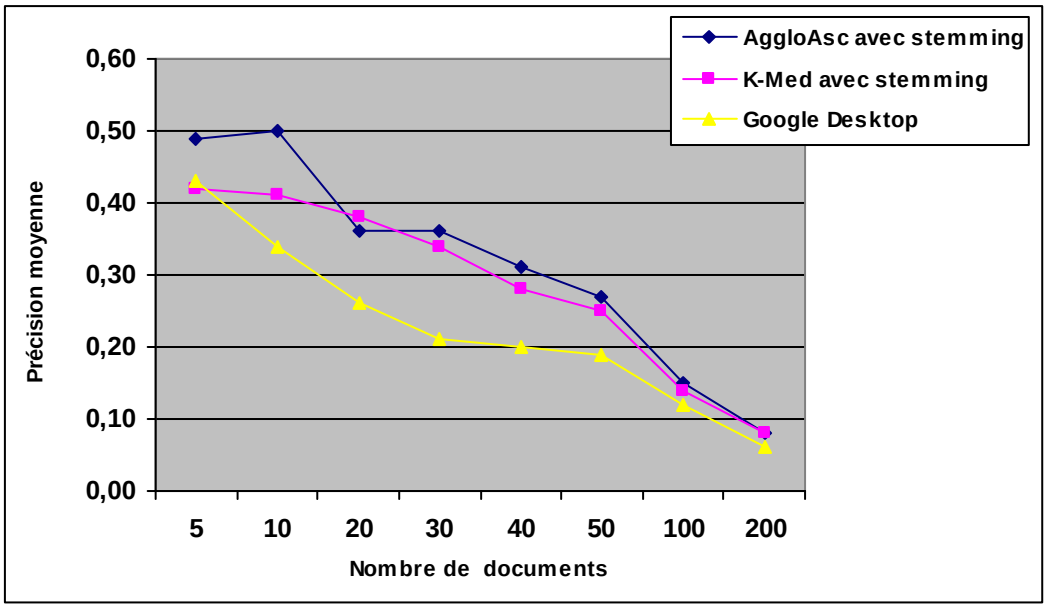


Figure 5.19 : Précision moyenne (partition en 13 clusters) des premiers documents retournés (avec stemming)

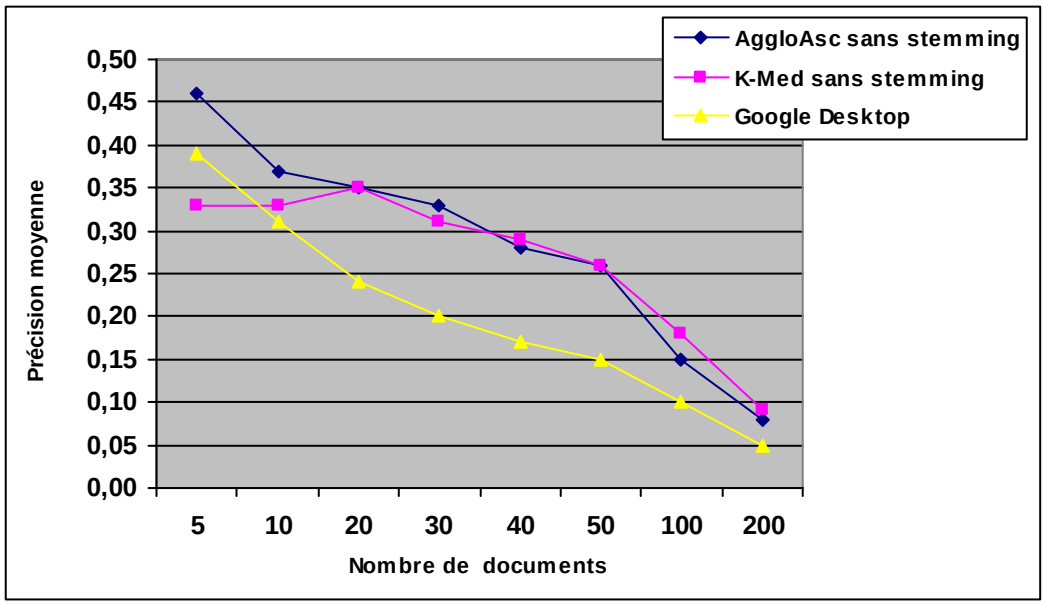


Figure 5.20 : Précision moyenne (partition en 13 clusters) des premiers documents retournés (sans stemming)

Rappel moyen

	à 5 documents retournés	à 10 documents retournés	à 20 documents retournés	à 30 documents retournés	A 40 documents retournés	à 50 documents retournés	à 100 documents retournés	à 200 documents retournés
AggloAsc avec stemming	0,15	0,32	0,43	0,63	0,72	0,80	0,90	0,98
K-Med avec stemming	0,13	0,26	0,45	0,62	0,68	0,74	0,84	1,00
Google Desktop	0,17	0,25	0,37	0,43	0,51	0,60	0,75	0,82
AggloAsc sans stemming	0,18	0,27	0,47	0,64	0,70	0,78	0,96	1,00
K-Med sans stemming	0,11	0,21	0,41	0,54	0,69	0,78	0,90	0,91
Google Desktop	0,15	0,23	0,33	0,40	0,45	0,49	0,58	0,63

Tableau 5.13 : Rappel moyen des premiers documents retournés

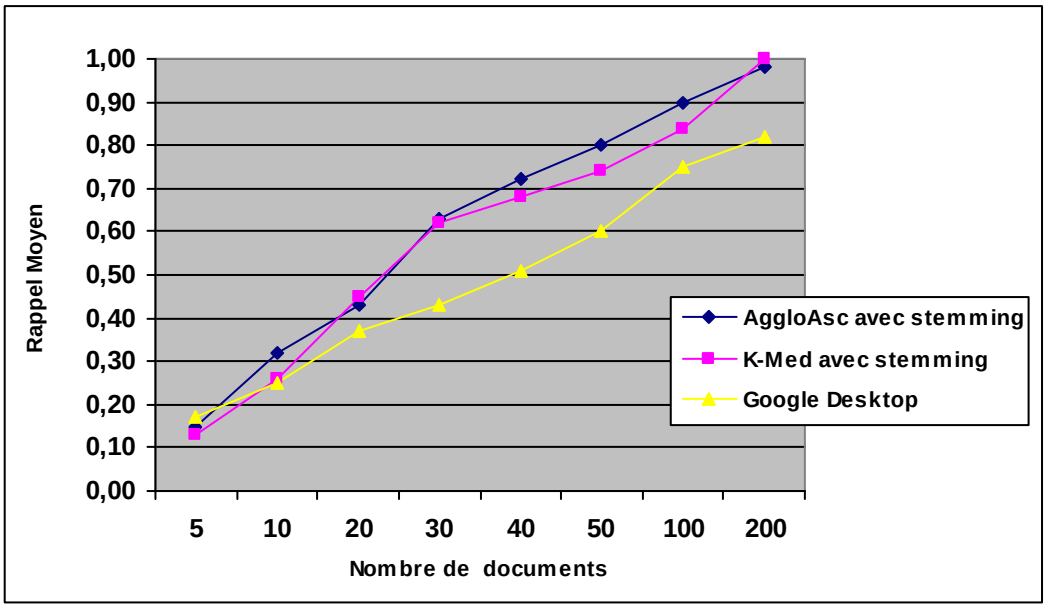


Figure 5.21 : Rappel moyen (partition en 13 clusters) des premiers documents retournés (avec stemming)

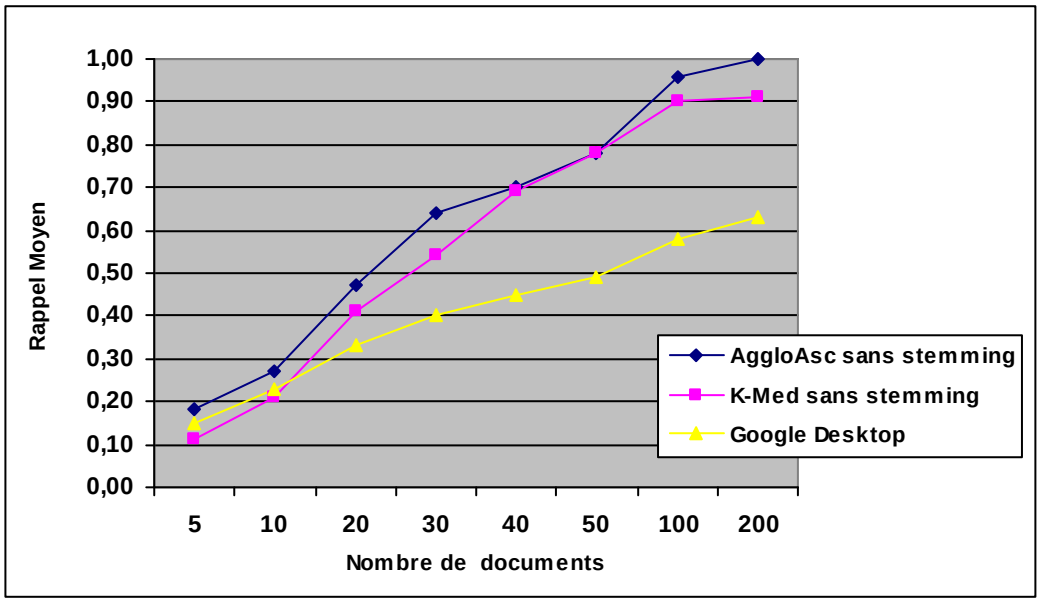


Figure 5.22 : Rappel moyen (partition en 13 clusters) des premiers documents retournés (sans stemming)

4.6. Performances de la l'approche proposée

Les performances décrites dans les sections 4.5.1 et 4.5.2, synthétisées dans les tableaux 5.14 et 5.15, réalisées par l'approche proposée (que ce soit dans le cas d'un stemming ou non) sont dues, essentiellement, au processus de clustering amélioré par un prétraitement de stemming. En effet, avec un clustering du corpus à l'état brut, les documents se rapportant à la même thématique, ont une forte probabilité de se retrouver dans le même cluster. Avec le stemming, cette probabilité augmente, donc, si un représentant d'un cluster est pertinent pour une requête, son voisinage le sera aussi (selon la stratégie décrite dans la

section 2, chapitre 5), puisque ce voisinage est construit en faisant intervenir l'ensemble des attributs des représentants des documents (section 4.2.2). Cela augmente les chances d'avoir plus de documents pertinents en tête de la liste rapportée, ceci est à l'inverse d'un système de recherche classique qui, quant à lui, ne fait intervenir que les mots contenus dans des requêtes, limitant ainsi l'espace de recherche des documents pertinents.

Il faut aussi noter l'importance du nombre de clusters qui est, comme nous l'avons vu, un facteur très important dans notre approche. En effet, nous avons remarqué qu'avec une valeur élevée de ce nombre, les résultats sont jugés satisfaisants, ceci est expliqué par la «**spécialisation des clusters»,** c'est-à-dire, qu'avec l'augmentation de ce nombre des thématiques qui se distinguent plus par des clusters disjoints, ce qui rend le rapprochement avec une requête plus facile et plus fructueux.

5. Conclusion

Dans ce chapitre, nous avons exposé les différentes étapes des expérimentations menées, afin d'évaluer la pertinence de l'approche proposée, exposée dans le chapitre 4, et son influence sur une recherche documentaire.

Avant de passer à l'évaluation de l'approche proposée, nous avons pu mettre en évidence l'importance d'un prétraitement tel que le stemming et son influence sur la qualité d'une classification non supervisée documentaire, ainsi que sur l'amélioration d'une recherche classique.

Dans une seconde étape, nous avons pu démontrer l'apport de l'approche proposée sur une recherche documentaire par rapport à une recherche classique avec et sans utilisation du stemming (tableaux récapitulatifs 5.14 et 5.15). Nous avons testé, avec succès, cette approche qui, en choisissant le bon nombre de clusters, a prouvé son efficacité en améliorant la précision et le rappel.

Nbr documents retournés	5			10			20			30			40			50			100			200		
Nbr Clusters	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**	**5**	**7**	**9**
Avec Stemming																								
AggloAsc																								
Rappel	0,07	0,07	0,11	0,14	0,12	0,16	0,22	0,18	0,24	0,30	0,21	0,32	0,37	0,29	0,43	0,49	0,38	0,52	0,84	0,94	0,81	0,92	0,94	1,00
Précision	0,21	0,19	0,25	0,22	0,18	0,21	0,19	0,15	0,18	0,18	0,13	0,17	0,16	0,13	0,18	0,16	0,13	0,17	0,14	0,16	0,14	0,08	0,08	0,08
K-Med																								
Rappel	0,11	0,12	0,08	0,17	0,21	0,13	0,27	0,29	0,25	0,38	0,36	0,38	0,46	0,41	0,51	0,52	0,52	0,59	0,75	0,76	0,78	0,92	0,94	0,93
Précision	0,30	0,24	0,23	0,25	0,23	0,21	0,22	0,20	0,20	0,21	0,17	0,21	0,19	0,16	0,21	0,18	0,17	0,19	0,12	0,12	0,12	0,08	0,08	0,07
Google Desktop																								
Rappel	0,17			0,25			0,37			0,43			0,51			0,60			0,75			0,82		
Précision	0,43			0,34			0,26			0,21			0,20			0,19			0,12			0,06		
Sans Stemming																								
AggloAsc																								
Rappel	0,03	0,06	0,07	0,07	0,11	0,13	0,14	0,18	0,23	0,17	0,24	0,34	0,23	0,29	0,43	0,31	0,37	0,51	0,64	0,65	0,73	0,83	1,00	0,99
Précision	0,11	0,15	0,21	0,13	0,16	0,21	0,13	0,16	0,20	0,11	0,16	0,21	0,11	0,14	0,19	0,12	0,14	0,18	0,12	0,12	0,13	0,07	0,08	0,08
K-Med																								
Rappel	0,04	0,06	0,10	0,09	0,10	0,20	0,15	0,16	0,31	0,20	0,24	0,47	0,23	0,31	0,53	0,30	0,38	0,62	0,54	0,62	0,94	0,98	0,92	1,00
Précision	0,14	0,15	0,29	0,14	0,14	0,27	0,13	0,12	0,22	0,12	0,13	0,23	0,11	0,13	0,20	0,12	0,12	0,19	0,09	0,10	0,15	0,08	0,07	0,08
Google Desktop																								
Rappel	0,15			0,23			0,33			0,40			0,45			0,49			0,58			0,63		
Précision	0,39			0,31			0,24			0,20			0,17			0,15			0,10			0,05		

Tableau 5.14 : Précision et Rappel moyens de l'approche proposée et du moteur de recherche classique avec 5, 7 et 9 clusters

Nbr documents retournés	5			10			20			30			40			50			100			200		
Nbr Clusters	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**	**11**	**13**	**15**
Avec Stemming																								
AggloAsc																								
Rappel	0,14	0,15	0,16	0,27	0,32	0,31	0,39	0,43	0,42	0,57	0,63	0,62	0,65	0,72	0,70	0,70	0,80	0,77	0,85	0,90	0,88	0,98	0,98	0,98
Précision	0,43	0,49	0,49	0,44	0,50	0,48	0,34	0,36	0,34	0,34	0,36	0,34	0,29	0,31	0,30	0,25	0,27	0,27	0,15	0,15	0,15	0,08	0,08	0,08
K-Med																								
Rappel	0,14	0,13	0,12	0,27	0,26	0,25	0,44	0,45	0,38	0,60	0,62	0,57	0,70	0,68	0,67	0,77	0,74	0,76	0,92	0,84	0,85	1,00	1,00	1,00
Précision	0,42	0,42	0,35	0,40	0,41	0,36	0,36	0,38	0,31	0,33	0,34	0,32	0,29	0,28	0,28	0,26	0,25	0,26	0,15	0,14	0,15	0,08	0,08	0,08
Google Desktop																								
Rappel	0,17			0,25			0,37			0,43			0,51			0,60			0,75			0,82		
Précision	0,43			0,34			0,26			0,21			0,20			0,19			0,12			0,06		
Sans Stemming																								
AggloAsc																								
Rappel	0,16	0,18	0,13	0,25	0,27	0,25	0,43	0,47	0,43	0,57	0,64	0,60	0,64	0,70	0,67	0,71	0,78	0,75	0,81	0,96	0,96	0,88	1,00	1,00
Précision	0,40	0,46	0,40	0,34	0,37	0,38	0,32	0,35	0,34	0,30	0,33	0,33	0,26	0,28	0,28	0,24	0,26	0,26	0,14	0,15	0,16	0,07	0,08	0,08
K-Med																								
Rappel	0,14	0,11	0,11	0,25	0,21	0,23	0,47	0,41	0,37	0,64	0,54	0,57	0,75	0,69	0,67	0,82	0,78	0,76	0,92	0,90	0,90	0,93	0,91	0,92
Précision	0,36	0,33	0,35	0,34	0,33	0,35	0,35	0,35	0,31	0,33	0,31	0,32	0,29	0,29	0,28	0,26	0,26	0,26	0,15	0,18	0,17	0,09	0,09	0,09
Google Desktop																								
Rappel	0,15			0,23			0,33			0,40			0,45			0,49			0,58			0,63		
Précision	0,39			0,31			0,24			0,20			0,17			0,15			0,10			0,05		

Tableau 5.15 : Précision et Rappel moyens de l'approche proposée et du moteur de recherche classique avec 11, 13 et 15 clusters

Conclusion et Perspectives

Les objectifs que nous nous sommes fixés (évaluation de l'influence de la nature de la langue arabe sur la fiabilité d'une approche de recherche documentaire basée sur une classification non supervisée avec prise en compte de deux paramètres essentiels, à savoir un traitement de radicalisation ou «stemming» et le nombre de clusters utilisé) ont été atteints. Pour cela, des techniques qui ont été utilisées (modèle vectoriel, pondération TF-IDF, mesure cosinus, méthode de classification par agglomération ascendante et méthode des K-médoïds) sont des techniques qui ont été déjà testés et ont prouvé leurs efficacités dans ce domaine.

Nous avons constaté en premier lieu, que la langue arabe réagie bien au stemming dans la génération des clusters qui sont d'une qualité meilleure dans le cas de stemming que ceux dans l'état brut. Cette constatation est renforcée par l'amélioration de la précision et du rappel dans une recherche documentaire classique, ce qui est en parfait accord avec des travaux précurseurs. Ces améliorations sont dues au fait que le stemming ou la radicalisation atténue le caractère flexionnel de la langue arabe malgré les ambiguïtés qu'il peut causer dans certains cas. Ces cas sont, à notre avis et vu les résultats obtenus, couverts par le fort taux des stems **corrects** extraits après un traitement de stemming**.**

L'amélioration qu'une classification non supervisée, effectuée avec deux méthodes de clustering, peut apporter à une recherche documentaire sur des textes à l'état brut et encore mieux sur des textes stemmés ou radicalisés a été démontrée. Ceci est appuyé par la nette amélioration de la précision et du rappel. Les expérimentations ont été effectuées avec plusieurs partitions en clusters et dont les résultats ont été différents, ce qui reflète l'influence du choix du nombre de clusters.

Notre travail s'est déroulé sur un corpus relativement petit, il reste a expérimenté l'approche proposée sur un corpus de taille plus grande, qui à notre avis, demandera plus de techniques pour le choix du nombre de clusters et des documents à retourner, ceci étant donné que les clusters seront plus peuplés que ceux dans notre cas.

Nous n'avons pas pu tester l'influence de la réduction de la dimension des vecteurs de représentation, toujours en langue arabe, sur la qualité d'une classification non supervisée dans la recherche documentaire, ce qui reste dans nos perspectives les plus proches.

Il faut aussi réfléchir sur la façon d'adapter l'approche proposée sur la liste des documents retournée par un moteur de recherche sur le Net, où seule la partie textuelle de ces documents est utilisée dans le processus de clustering.

La classification non supervisée **«descriptive»** des textes en langue arabe reste un autre axe ouvert aux recherches. Les clusters produits par une classification non supervisée n'étant pas labellisés ou labellisés manuellement, il faut réfléchir sur le moyen de les décrire automatiquement.

Références et Bibliographie

1. [Abdelali et al., 2004a]; Ahmed Abdelali, Jim Cowie and Hamdy S. Soliman: «**Arabic Information Retrieval Perspectives»,** JEP-TALN 2004, Arabic language processing, 19-22 April, Fez, Morocco, 2004.

2. [Abdelali et al., 2004b]; Ahmed Abdelali, Jim Cowie and Hamdy S. Soliman: «**Building A Modern Standard Arabic Corpus»,** Computing Research Laboratory, New Mexico State University, USA, 2004.

3. [Abdelali et Cowie, 2004]; Ahmed Abdelali and Jim Cowie: « **Regional Corpus of Modern Standard Arabic»,** Computing Research Laboratory, New Mexico State University, USA, 2004.

4. [Ah-Pine et al., 2005]**;** Julien Ah-Pine, Julien Lemoine et Hamid Benhadda: **«Un nouvel outil de classification non supervisée de documents pour la découverte de connaissances et la détection de signaux faibles : rares text »,** île Rousse 2005, Journée sur les systèmes d'information élaborée, France, 2005.

5. [Aljlayl et Frieder, 2002]**;** M. Aljlayl and O. Frieder: «**On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach**», 11 the International Conference on Information and Knowledge Management (CIKM), November, Virginia, USA, 2002.

6. [Al-Kharashi et Evens, 1994]**;** J. Al-Kharashi and M. W. Evens: «**Comparing words, stems and roots as index terms in an Arabic information retrieval system**», Journal of the American Society for Information Science (JASIS) 45 (8), USA, 1994.

7. [Attia, 2000]; M. Attia: «**A large-scale computational processor of the Arabic morphology**», A Master's Thesis, Cairo University, Egypt, 2000.

8. [Baloul et al., 2002]**;** S. Baloul, M. Alissali, M. Baudry et P. Boula de Mareüil: «**Interface syntaxe-prosodie dans un système de synthèse de la parole à partir du texte en arabe** », 24es Journées d'Étude sur la Parole, 24-27 juin, Nancy, France, 2002.

9. [Baldi et al., 2003]**;** P. Baldi, P. Frasconi and P. Smyth: «**Modeling the Internet and the Web: Probabilistic Methods and Algorithms»,** ISBN: 0-470-84906-1, USA, 2003.

10. [Bargeton et Devèze, 2005]**;** Alexandre Bargeton et Benjamin Devèze: «**Comparaison de différentes techniques d'optimisation pour l'apprentissage non-supervisé »,** Master IAD, rapport de projet de recherche (PRREC), Université Pierre et Marie Curie, France, 2005.

11. [Bellot, 2000]; Patrice Bellot: «**Méthodes de classification et de segmentation locales non supervisées pour la recherche documentaire»,** Thèse de Doctorat, Université d'Avignon et des Pays de Vaucluse, France, 2000.

12. [Bellot et El-bèze, 2000]; Patrice Bellot et Marc El-Bèze: **«Classification locale non supervisée pour la recherche documentaire** », Traitement Automatique des Langues (TAL), vol. 42, n°2, janvier, édition Hermès, 2001.

13. [Berkhin, 2002]**;** Pavel Berkhin: «**Survey of Clustering Data Mining Techniques**», Accrue Software, CA, USA, 2002.

14. [Boulaknadel, 2005]**;** Siham Boulaknadel: «**Utilisation des syntagmes nominaux dans un système de recherche d'information en langue arabe** », LINA FRE CNRS 2729-Université de Nantes 2, France, 2005.

15. [Brill, 1994]**;** Eric Brill: «**Advances in Transformation Based Part of Speech Tagging**», Proceedings of the Twelfth International Conference on Artificial Intelligence (AAAI-94), Seattle, WA, USA, 1994.

16. [Buckwalter, 2002]; T. Buckwalter: «**Buckwalter Morphological Analyzer Version 1.0**», http://www.qamus.org

17. [Candillier et al., 2005]; Laurent Candillier, Isabelle Tellier et Fabien Torre1: «**Tuareg: Classification non supervisée contextualisée »,** Université Charles de Gaulle - Lille 3, France, 2005.

18. [Caropreso et al., 2001]**;** M. F. Caropreso, S. Matwin, and F. Sebastiani: «**A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization** ». In Chin, A. G., editor, Text Databases and Document Management: Theory and Practice, pages 78–102. Idea Group Publishing, Hershey, USA, 2001.

19. [Chauché et al., 2003]**;** Jacques Chauché, Violaine Prince, Simon Jaillet et Maguelonne Teisseire: « **Classification automatique de textes à partir de leur analyse syntaxico-sémantique »,** TALN, Batz-sur-Mer, France, 2003.

20. [Chavent et al., 1999]**;** Marie Chavent, Christiane Guinot, Yves Lechevallier et Michel Tenenhaus: «**Méthodes divisives de classification et segmentation non supervisée: recherche d'une typologie de la peau humaine saine»,** Revue de statistique appliquée, tome 47, n°4, p.87-99, France, 1999.

21. [CIEP, 2007]**;** Centre International des études pédagogiques http://www.ciep.fr/publications/genetique/genetique31.php

22. [Darwish, 2002]**;** K. Darwish: «**Building a shallow morphological analyzer in one day**», ACL Workshop on Computational Approaches to Semitic languages, July 11, Philadephia, Pennsylvania, USA, 2002.

23. [Darwish et Oard, 2002]**;** K. Darwish and D. W. Oard: «**CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval**», TREC, Gaithersburg: NIST, pp 703-710, USA, 2002.

24. [Darwish et al., 2005]**;** Kareem Darwish, Hany Hassan and Ossama Emam: « **Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval**», Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 25–30, June, Ann Arbor, USA, 2005.

25. [De Loupy, 2000]**;** C. de Loupy: «**Évaluation de l'apport de connaissances linguistiques en désambiguïsation sémantique et recherche documentaire**», Thèse de doctorat, Laboratoire informatique d'Avignon, université d'Avignon et des Pays de Vaucluse, France, 2000.

26. [Deerwester et al., 1990]**;** S. Deerwester, S. Dumais, T. Landauer, G. Furnas and R. Harshman: « **Indexing by latent semantic analysis**», Journal of the American Society of Information science, (JASIS) 416(6): 391–407, USA, 1990.

27. [Débili et al., 2002]; F. Débili, H. Achour H. et E. Souici: «**La langue arabe et l'ordinateur: de l'étiquetage grammatical à la voyellation automatique** », Correspondances de l'IRMC, N° 71, pp. 10-28, juillet-août, France, 2002.

28. [Denoyer, 2004]**;** Ludovic Denoyer: «**Apprentissage et inférence statistique dans les bases de documents structurés : Application aux corpus de documents textuels** », Thèse de Doctorat, Université Paris 6, France, 2004.

29. [Diab et al., 2004]**;** Mona Diab, Kadri Hacioglu and Daniel Jurafsky: «**Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks** », Stanford University, USA, 2004.

30. [Diday, 1971]**;** E. Diday: «**Une nouvelle méthode de classification et reconnaissance des formes la méthode des nuées dynamiques**», Revue de statistique appliquée, tome 19 n°:2, p. 19-33. Société française des statistiques, France, 1971.

31. [Douzidia, 2004]**;** Fouad Soufiane Douzidia: «**Résumé automatique de texte arabe »,** Université de Montréal, Canada, 2004.

32. [Dubois, 2002]; Julien Duboi: « **Classification automatique de courrier électronique** », Université de Montréal, Canada, 2002.

33. [El Kassas, 2005]**;** Dina El Kassas : «**Une étude contrastive de l'arabe et du français dans une perspective de génération multilingue»,** UFR Linguistique, Université Paris 7 – Denis Diderot, France, 2005.

34. [Faber, 1994]; Vance Faber: «**Clustering and the Continuous k-Means Algorithm**», Los Alamos Science Number 22, USA, 1994.

35. [Fegas, 2005]; Mounir Fegas: « **Classification de documents XML, Application au corpus d'INEX et aux rapports d'activité INRIA** », LRI Université de Paris Sud XI, France, 2005.

36. [Fuhr et Buckley, 1991]; N. Fuhr and C. Buckley: «**A probabilistic learning approach for document indexing**». In ACM Transactions on Information Systems, volume 9, pages 223–248, France, 1991.

37. [Gallinari et al., 1999]**;** Patrick Gallinari, Hugo Zaragoza et Massih Amini: «**Apprentissage et données textuelles »,** LIP6, Université Paris 6, France, 1999.

38. [Genane, 2004]; Genane Youness: «**Contributions à une méthodologie de comparaison de partitions»,** Thèse de Doctorat, Université Paris 6, France, 2004.

39. [Goweder et De Roeck, 2001]; Abduelbaset Goweder et Anne De Roeck: «**Assessment of a Significant Arabic Corpus**», University of Essex, England, 2001.

40. [Huot et Coupet, 2005]; Charles Huot, Pascal Coupet: «**Le Text Mining sur la langue Arabe : application au traitement des sources ouvertes**», TEMIS SA, Paris, France, 2005.

41. [Ignat et Rousselot, 2006]**;** Camelia Ignat et François Rousselot: «**Un algorithme de génération de profil de document et on évaluation dans le contexte de la classification thématique »,** JADT: 8emes Journées internationales d'Analyse statistique des Données Textuelles, France, 2006.

42. [Jaillet, 2003]; Simon Jaillet: «**Catégorisation automatique de documents** », LIRMM, Montpellier, France, 2003.

43. [Jaillet et al., 2003]; Simon Jaillet, Maguelonne Teisseire, Gérard Dray: «**Adéquation des modèles de représentation aux méthodes de catégorisation »**, LIRMM, Montpellier, France, 2003.

44. [Jain et al., 2000]; A.K. Jain, M.N. Murty, P.J. Flynn: « **Data Clustering: A Review »,** Michigan State University, USA, 2000.

45. [Jalam, 2003]; Radwan JALAM: « **Apprentissage automatique et catégorisation de textes multilingues »,** Université Lumière Lyon 2, France, 2003.

46. [Jardino, 2004]; Michèle Jardino: **« Recherche de structures latentes dans des partitions de textes de 2 à K classes »,** LIMSI – CNRS, France, 2004.

47. [Jardino, 2005]; Michèle Jardino: «**Fouille de données dans les corpus de textes**», http://www.limsi.fr/Recherche/LIR

48. [Kanungo et al., 2002]; Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman and Angela Y. Wu: «**An Efficient k-Means Clustering Algorithm: Analysis and Implementation**», IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 24, NO. 7, July, USA, 2002.

49. [Kelaiaia et al., 2007a]; Abdessalem Kelaiaia, Hayet Farida Merouani et Yamina Tlili-Guiassa: **«Classification non supervisée de textes en langue arabe»,** Proceedings of JED, Journées Ecole Doctorale & Réseaux de Recherche en Sciences et Technologies de l'Information, 27-28 Mai 2007, Annaba, Algérie, 2007.

50. [Kelaiaia et al., 2007b]; Abdessalem Kelaiaia, Hayet Farida Merouani et Yamina Tlili-Guiassa: **«Application de la méthode des k-moyennes hiérarchique sur un corpus de textes arabes»,** Proceedings of COSI'2007, Quatrième Colloque sur l'Optimisation et les systèmes d'Information, pp. 327-335, 11-13 Juin, Oran, Algérie, 2007.

51. [Khalis, 2006]; Zohra KHALIS : **« La segmentation thématique, Application à la campagne DEFT'06»**, CLIPS – MRIM, France, 2006.

52. [Khoja, 2001]; Shereen Khoja : «**APT: Arabic Part-of-speech Tagger**», Actes de l'atelier des étudiants de Second Meeting of the North American Chapter of the Association for Computational Linguistics, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2001.

53. [Kiraz, 1996]; G. A. Kiraz: **«Analysis of the Arabic Broken Plural and Diminutive**», In Proceedings of the 5th International Conference and Exhibition on Multi-Lingual Computing, Cambridge, UK, 1996.

54. [Korenius et al., 2004]; Tuomo Korenius, Jorma Laurikkala, Kalervo Järvelin and Martti Juhola: « **Stemming and Lemmatization in the Clustering of Finnish Text Documents**», Department of Computer Sciences, Center for Advanced Studies, FIN-33014 University of Tampere, Finland, 2004.

55. [Larkey et al., 2002]; Leah S. Larkey, Lisa Ballesteros and Margaret E. Connell: «**Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis** », la 25ème édition de la conférence de la recherche et de développement en recherche d'information, SIGIR, Tampere, Finland, 2002.

56. [Larkey et al., 2005]; Leah S. Larkey, Lisa Ballesteros and Margaret E. Connell: «**Light Stemming for Arabic Information Retrieval**», Univ. of Massachusetts, Dept. of Computer Science, USA, 2005.

57. [LDC, 2007]; Linguistic Data Consortium, University of Pennsylvania, USA. 2007. http://www.ldc.upenn.edu

58. [Leclerc, 2000]; J. Leclerc: «**L'aménagement linguistique dans le monde**», http://www.tlfq.ulaval.ca/axl/monde/famarabe.htm

59. [Lemur, 2006]; The Lemur Toolkit for Language Modeling and Information Retrieval, http://www.lemurproject.org

60. [Lewis, 1992]; D. Lewis: «**Representation and learning in information retrieval**», PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, USA, 1992.

61. [Maamouri et Bies, 2004]; Mohamed Maamouri and Ann Bies : «**Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools** », LDC, University of Pennsylvania, Philadelphia, PA 19104, USA, 2004.

62. [Mladenic et Grobelnik, 2003]; Dunja MLADENIĆ, Marko GROBELNIK: «**Text and web mining**», Kluwer Academic Publishers, USA, 2003.

63. [Memmi, 2001]; D. Memmi: **« Le modèle vectoriel pour le traitement de documents ».** UQAM**,** Canada, 2001.

64. [Miller et al., 1999]; E. Miller, D. Shen, J. Liu et C. Nicholas: «**Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System** », Journal of Digital Information, USA, 1999.

65. [NEMLAR, 2005]; **NEMLAR's homepage**, http://www.nemlar.org

66. [Ng et Han, 1994]; R.T. Ng and J. Han: «**Efficient and Effective Clustering Methods for Spatial Data Mining »**, Conference on Very Large Databases, USA, 1994.

67. [Pasquier, 2003]; Nicolas Pasquier: « **Fouille de Données: Classification non supervisée »,** Université de Nice Sophia-Antipolis Laboratoire I3S, France, 2003.

68. [Porter, 1980]; Martin F. Porter: «**An Algorithm for Suffix Stripping»,** http://tartarus.org/~martin/PorterStemmer/def.txt

69. [Preux, 2006]; Philippe Preux : « **Fouille de données** », université de Lille 3, France, 2006.

70. [Rijsbergen, 1979]; Keith van Rijsbergen: «**Information Retrieval: Uncertainty and Logics: Advanced Models for the representation and retrieval systems**», http://books.google.fr

71. [Salton et Buckley, 1988]; Gerard Salton and Christopher Buckley: « **Term-weighting approaches in automatic text retrieval** », Department of Computer Science, Cornell University, Ithaca, NY 14853, USA, 1988.

72. [Sawaf et al., 2001]; Hassan Sawaf, Jörg Zaplo and Hermann Ney: «**Statistical Classification Methods for Arabic News Articles** », AIXPLAIN AG Monnetstrasse 18 D-52146 Würselen, Germany, 2001.

73. [Sebastiani, 1999]; F. Sebastiani: «**A Tutorial on Automated Text Categorisation**», Proceedings of ASAI, 1st Argentinian Symposium on Artificial Intelligence, pp. 7-35, Argentine, 1999.

74. [Sebastiani, 2002]; Fabrizio Sebastiani: «**Machine Learning in Automated Text Categorization**», Consiglio Nazionale delle Ricerche, Italy, 2002.

75. [Schmid, 1994]; H. Schmid: « **Probabilistic part-of-speech tagging using decision trees**». International Conference on New Methods in Language Processing, Manchester, UK. 1994.

76. [Steinbach et al., 1999]; Michael Steinbach, George Karypis and Vipin Kumar**: «A Comparison of Document Clustering Techniques »,** Department of Computer Science and Engineering, University of Minnesota, USA, 1999.

77. [Sulaiti, 2003]; Latifa El Sulaiti: «**L'arabe contemporain**», Radio Qatar, Qatar, 2003.

78. [Sulaiti, 2005]; **Latifa Al-Sulaiti's homepage**, http://www.comp.leeds.ac.uk/eric/latifa/index.htm

79. [Tuerlinckx, 2004]; Laurence Tuerlinckx: «**La lemmatisation de l'arabe non classique**», JADT: 7es Journées internationales d'Analyse statistique des Données Textuelle, France, 2004.

80. [Turenne, 2000]; Nicolas Turenne: **« Apprentissage statistique pour l'extraction de concepts à partir du textes. Application au filtrage d'information textuelle»,** Université Louis Pasteur, Starsbourg, France, 2000.

81. [Weiss, 2006]; Dawid Weiss: «**Descriptive Clustering as a Method for Exploring Text Collections**», PhD thesis, Institute of Computing Science, Poznàn, Poland. 2006.

82. [Witte, 2006]; René Witte: «**Introduction to Text Mining**», EDBT, Faculty of Informatics, Institute for Program Structures and Data Organization (IPD), Université Karlsruhe, Germany, 2006.

83. [Xerox, 1997]; Xerox Arabic Home Page http://www.cis.upenn.edu/~cis639/arabic/home.html

84. [Xu et al., 2002]; J. Xu, A. Fraser and R. Weischedel: «**Empirical Studies in Strategies for Arabic Retrieval**», Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR), pp. 269-274. August 11-15, Finland, 2002.

85. [Yang et Pederson, 1997]; Y. Yang and J.O. Pedersen: «**A Comparative Study on Feature Selection in Text Categorization**», Proceedings of ICML, 14th International Conference on Machine Learning, pp. 412-420, Morgan Kaufmann, San Francisco, USA, 1997.

86. [Zhai, 2002]; Cheng Xiang Zhai: «**Risk Minimization and Language Modeling in Text Retrieval**», School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 2002.

Annexes

<u>Annexe 1</u> : Liste des requêtes (Brutes, Translittérées et stemmées) avec le nombre et réponses pertinentes

Q01 ما هو مستوى السياحة في بلدان الخليج العربي (19)
Mstwy AlsyAHp bldAn Alxlyj AlErby
Mstw syAH bld xlyj Erb

Liste des documents pertinents
To01, To02, To05,To11, To13, To14, To20, To27, To28, To29, To31, To34, To35, To36, To37, To39, To40, To41, To42

Q02 ما هو مستوى السياحة في البلدان العربية (35)
Mstwy AlsyAHp AlbldAn AlErbyp
Mstw syAH bld Erb

Liste des documents pertinents
To01, To02, To04, To05, To08, To11, To13, To14, To20, To21, To26, To27, To28, To29, To30, To31, To32, To33, To35, To36, To37, To39, To40, To41, To42, To43, To44, To45, To46 , To47, To48, To49, To50, To52, To55

Q03 ما هي العلاقات التي قد توجد بين السياحة و عالم الطيران و السفر الجوي (20)
AlElAqAt Alty twjd AlsyAHp EAlm AlTyrAn Alsfr Aljwy
ElAq twjd syAH EAlm Tyr sfr jw

Liste des documents pertinents
To03, To04, To05, To07, To10, To11, To12, To17, To18, To22, To30, To34, To36, To39, To40, To42, To45, To48, To54, To56

Q04 ما هو مستوى تأثيرالسياحة عبر بلدان العالم (31)
Mstwy tAvyr AlsyAHp Ebr bldAn AlEAlm
Mstw tAvyr syAH Ebr bld EAlm

Liste des documents pertinents
To01, To02, To06, To09, To14, To15, To16, To17, To18, To19, To21, To26, To27, To31, To35, To36, To37, To40, To41, To42, To43, To44, To45, To46, To47, To48, To49, To50, To52, To53, To55

Q05 أشهر الفنادق العالمية و المنتجعات (22)
APhr AlfnAdq AlEAlmyp AlmntjEAt
APhr fnAdq EAlm mntjE

Liste des documents pertinents
To04, To06, To09, To14, To15, To16, To18, To19, To26, To27, To32, To35, To37, To39, To41, To43, To44, To46, To47, To48, To52, To55

Q06 مواسم الحج و العمرة (9)
mwAsm AlHj AlEmrp
mwAsm Hj Emr

Liste des documents pertinents
To12, To13, To20, To24, To27, To37, To39, To40, To49

Q07 الطيران في السعودية و البلدان العربية (15)
AlTyrAn AlsEwdyp AlbldAn AlErbyp
Tyr sEwd bld Erb

Liste des documents pertinents
To03, To04, To05, To07, To10, To11, To17, To22, To30, To34, To36, To42, To54, To56, To 57

Q08 ما هي الامراض التنفسية و كيف يتم معالجتها (7)
AlAmrAD Altnfsyp ytm mEAljthA
AmrAD tnfs mEAljt

Liste des documents pertinents
Hm01, Hm04, Hm08, Hm10, Hm12, Hm14, Hm20

Q09 ما هي الاساليب الناجعة للوقاية من الامراض (19)
AlAsAlyb AlnAjEp llwqAyp AlAmrAD
AsAlyb nAjE wqA AmrAD

Liste des documents pertinents
Hm02, Hm04, Hm06, Hm08, Hm09, Hm10, Hm12, Hm13, Hm14, Hm16, Hm19, Hm20, Hm21, Hm22, Hm25, Hm27, Hm28, Hm31, Hm32

Q10 الادوية الناجعة لمعالجة بعض الامراض و الحفاض على صحة الانسان (22)
AlAdwyp AlnAjEp lmEAljp bED AlAmrAD AlHfAD SHp AlAnsAn
Adw nAjE EAlj bED AmrAD HfAD SH Ans

Liste des documents pertinents
Hm01, Hm02, Hm03, Hm04, Hm08, Hm10, Hm12, Hm13, Hm14, Hm16, Hm20, Hm21, Hm22, Hm23, Hm24, Hm25, Hm27, Hm28, Hm29, Hm30, Hm31, Hm32

Q11 ما هي الاسباب المباشرة لظهور الامراض (21)
AlAsbAb AlmbAPrp lZhwr AlAmrAD
AsbAb mbAPr lZhwr AmrAD

Liste des documents pertinents
Hm01, Hm02, Hm04, Hm05, Hm06, Hm07, Hm08, Hm10, Hm12, Hm13, Hm14, Hm16, Hm17, Hm18, Hm19, Hm20, Hm21, Hm25, Hm27, Hm28, Hm32

Q12 ما هي المشاكل الصحية في العصر الحاضر و الاساليب للوقاية منها (23)
AlmPAkl AlSHyp AlESr AlHADr AlAsAlyb llwqAyp mnhA
mPAkl SH ESr HADr AsAlyb wqA mn

Liste des documents pertinents
Hm01, Hm03, Hm04, Hm06, Hm07, Hm08, Hm10, Hm12, Hm13, Hm14, Hm16, Hm17, Hm19, Hm20, Hm21, Hm22, Hm24, Hm25, Hm26, Hm27, Hm28, Hm30, Hm32

Q13 كيف يصاب الانسان بالامراض و ما هي مضاعفاتها (18)
ySAb AlAnsAn bAlAmrAD mDAEfAthA
ySAb Ans AmrAD mDAEfAt

Liste des documents pertinents
Hm01, Hm04, Hm06, Hm08, Hm10, Hm12, Hm13, Hm14, Hm16, Hm18, Hm19, Hm20, Hm21, Hm22, Hm25, Hm27, Hm28, Hm32

Q14 ما هي امراض العظام و كيف يصاب الانسان بها (7)
mAhy AmrAD AlEZAm ySAb AlAnsAn bhA
AmrAD EZAm ySAb Ans

Liste des documents pertinents
Hm06, Hm09, Hm16, Hm21, Hm22, Hm29, Hm31

Q15 البشرية و الرسالة الحضارية الاسلامية (18)
AlbPrypw AlrsAlp AlHDAryp AlAslAmyp
bPrypw rsAl HDAr AslAm

Liste des documents pertinents
Rel01, Rel02, Rel03, Rel04, Rel05, Rel06, Rel07, Rel08, Rel09, Rel10, Rel11, Rel12, Rel14, Rel15, Rel16, Rel17, Rel18, Rel19

Q16 الاسلام و مكوناته الاسلام دين و دنيا (8)
lAslAm mkwnAth dyn dnyA
slAm mkwnA dny

Liste des documents pertinents
Rel01, Rel03, Rel07, Rel08, Rel11, Rel15, Rel16, Rel18

Q17 عبر ومواعظ مستوحاة من عالم الحيوانات (18)
Ebr wmwAEZ mstwHAp EAlm AlHywAnAt
Ebr wAEZ mstwHA EAlm HywAn

Liste des documents pertinents
Chd02, Chd03, Chd04, Chd10, Chd11, Chd12, Chd13, Chd17, Chd18, Chd19, Chd20, Chd21, Chd22, Chd23, Chd24, Chd25, Chd26, Chd27

Q18 اللغة و الحضارة العربية و الاسلامية (6)
Allgp AlHDArp AlErbyp AlAslAmyp
Lg HDAr Erb AslAm

Liste des documents pertinents
Soc07, Soc13, Soc21, Soc24, Soc25, Rel15

Q19 البصل و الملح و البهار و الدجاج في الطبخ (9)
AlbSl AlmlH AlbhAr AldjAj AlTbx
bSl mlH bhAr djAj Tbx

Liste des documents pertinents
Rec01, Rec02, Rec03, Rec04, Rec05, Rec06, Rec07, Rec08, Rec09

Q20 الرياضة و كرة القدم في البلاد العربية (3)
AlryADp krp Alqdm AlblAd AlErbyp
ryAD kr qdm blad Erb

Liste des documents pertinents
Spo01, Spo02, Spo06

Q21 النزاع الفلسطيني الإسرائيلي و العربي الإسرائيلي (4)
AlnzAE AlflsTyny AlAsrAAyly AlErby AlAsrAAyly
nzAE flsTyn AsrAAyl Erb AsrAAyl

Liste des documents pertinents
Pol04, Pol06, Pol07, Pol08

<u>Annexe 2</u> : Liste des *stop words* utilisés dans *Al-Stem* (Fichier stoplist.txt)

Aly
byn
tHt
Ely
Ah
Al
Am
An
Ah
Aw
Al
AlA
ty
fy
qd
lqd
lA
mA
mE
hl
OA
hOA
hOh
hOAn
hAtyn
hAlAA
AllAty
AllAAy
AllwAty
tlk
AnA
nHn
Ant
AntmA
Antm
Antn
hw
hy
hmA
hm
hn
mA
mn
AynmA
mty
Ayn
AyAn
lmA
AOA
klmA
mhmA
AO
Hyv
HyvmA
Any
kyfmA
kyf
Hl
mn
fym
mA
Ayn
mty
Any
km
AyAn
bm
lm
mm
lmAOA
mAOA
AlA
Em
ElAm
AlAm
Emn
Ely
Aly
fy
mn
En
ky
w
W
mO
mnO
Hty
xlA
EdA
HAPA
vm
Aw
Am
bl
lkn
lA
Hty
yA
AyA
hyA
Ay
ln
lm
lmA
lA
mA
An
An
qd
lA
An
AO
lw
lwlA
AmA
lmA
hA
yA
AlA
hlA
An
mA
lw
ky
An
kAn
lkn
lyt
lEl
lA
Esy

<u>Annexe 3</u> : Liste des 50 premiers fichiers retournés par l'approche proposée en réponse à la requête 3

«ما هي العلاقات التي قد توجد بين السياحة و عالم الطيران و السفر الجوي»

To31_Stem.txt, To53_Stem.txt, To15_Stem.txt, To45_Stem.txt, To14_Stem.txt, To09_Stem.txt, To55_Stem.txt, To40_Stem.txt, To26_Stem.txt, To47_Stem.txt, To34_Stem.txt, To38_Stem.txt, To05_Stem.txt, To41_Stem.txt, To42_Stem.txt, To28_Stem.txt, To44_Stem.txt, To18_Stem.txt, To01_Stem.txt, To06_Stem.txt, To32_Stem.txt, To16_Stem.txt, To13_Stem.txt, To04_Stem.txt, To49_Stem.txt, To25_Stem.txt, To29_Stem.txt, To52_Stem.txt, To19_Stem.txt, To48_Stem.txt, To11_Stem.txt, To21_Stem.txt, To03_Stem.txt, To17_Stem.txt, To46_Stem.txt, To07_Stem.txt, To30_Stem.txt, To12_Stem.txt, To54_Stem.txt, To56_Stem.txt, To43_Stem.txt, To36_Stem.txt, To10_Stem.txt, To57_Stem.txt, AUT35_Stem.txt, To02_Stem.txt, Ec06_Stem.txt, To27_Stem.txt, To20_Stem.txt, Ec29_Stem.txt,