REPUBLIC ALGERIENNE DEMOCRATIQUEETPOPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE BLIDA USDB
FACULTE DES SCIENCES
DEPARTEMENT INFORMATIQUE

Mémoire de fin d'étude

En vue l'obtention du diplôme de Master 2 en informatique
OPTION : ingénierie du logiciel

THEME

Un Opérateur d'agrégation pour les données texte

Réalisé par :

- **M^lle^ GUERROUMI Nadia**
- **M^lle^ MAARICHE Asma**

Encadré par :

M^lle^ BEBBLIDIA Nadjia

M^lle^ OUKID Lamia

Soutenu le 23/09/2013, Devant le jury :

Président : M. FERFERA

Examinateur : M. NAHEL

Examinateur : M. HADJ YAHIA

Promotion
2012/2013

Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur : **M^lle^ OUKID Lamia**, pour ces conseils et sa disponibilité durant la période du travail.

Nous tenons à remercier particulièrement **M^lle^ BENBLIDIA Nadjia,** notre promotrice, pour ces conseils.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier notre parent pour le chaleureux soutien et aussi toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

DEDICACE

Je dédie se modeste travail à mes chers parents et ma grande mère.

A mes chers sœurs ***Aicha*** *et son marie Cherif et ses enfants* ***Sarah*** *et* ***Abdellatif*** *sans oublier le petit* ***Yasser*** *qui j'aime beaucoup, à* ***Saliha*** *qu'est très loin mais toujours dans mon cœur et son marie Omar et* ***Razika****, à mes frère et à toute ma grande famille.*

A mes amis spécialement Nadia, Imane, Hanane, Karima, Zinebe, Zainebe, Hadjere, Fatiha, Naima.

Sans oublier notre promotrice et co-promotricce.

ASMA

Dédicace

Avec une énorme joie, je dédie le fuit de ce travail à :

Ma chère mère, pour leur sacrifice, leur compréhension, leu soutien et tout ce qu'elle a fait pour ma réussite, que Dieu m'aidé à leur rendre le peut de ce qu'elle a fait pour mois.

Mes chèrs frères Ahmed et amine.

Ma sœur Nawel et son mari Fouad et sa Belle fille Souha

A tout ma famille

A tout mes amis : Asma, ma chère khadidja, Karima ……..

A notre Encadreur : M^lle OUKID Lamia

Nadia

Résumé :

Avec l'accroissement des données sur internet et dans les entreprises, la quantité des documents devient très importante et difficile à gérer par les décideurs. D'où la nécessité d'outils d'aide à la décision pour l'analyse de ce type de données. Les méthodes et les outils qu'offrent les technologies d'entreposage et d'analyse en ligne de données sont efficaces uniquement pour l'analyse de données numériques. Dans le même contexte, l'agrégation de données numériques s'effectue à l'aide d'opérateurs d'agrégation classiques : somme, moyenne, max, min, etc. Or ces opérateurs ne sont pas adaptés pour l'agrégation de données texte. Le but de ce travail est de définir un opérateur d'agrégation adapté à la nature des données texte (non structurées). Cela, en s'inspirant des techniques issues de la fouille de données.

Mots-clés : Analyse en ligne, Entrepôt de données, Cube OLAP, agrégation de données textuelles, fouille de données.

Abstract :

With increasing data on the internet and in business, the quantity of documents is very important and difficult to manage by policy makers. Hence the need of decision supports for the analysis of such data tools. The methods and tools offered by the storage and analysis of data online technologies are only effective for the analysis of numerical data. In the same context, the aggregation of digital data is done using conventional aggregation operators: sum, average, max, min, etc.. However, these operators are not suitable for text data aggregation.

The aim of this work is to define an aggregation operator adapted for text data, with drawing on techniques from data mining.

Keywords: OLAP, Data warehouse, text data aggregation, data mining.

الملخص

أدى التزايد الكثيف للبيانات و الأعمال التجارية على شبكة الأنترنيت و الكمية الهائلة للوثائق إلى صعوبة كبيرة بالنسبة لصانعي القرار. و لهذا كانت هناك حاجة ماسة لأدوات التحليل من أجل دعم هذا القرار وبالعودة إلى هذه الأدوات و كل التكنولوجيات الأخرى نجدها تطبق على البيانات الرقمية فقط وفي نفس السباق تعتبر هذه الأدوات تقليدية و لا تصلح لكي تطبق على البيانات النصية.

إذن هدفنا واضح و هو إيجاد تقنية نستطيع تطبيقها على البيانات النصية لاستخراج القرار الصائب.

الكلمات الخاصة : أدوات التحليل , البيانات الرقمية, البيانات النصية.

La listes des Figures :

TABLE DES MATIERES

Chapitre III : Etude sur les méthodes de fouille de texte

Parie II : *Conception et Réalisation*

Chapitre IV : Conception

Introduction Générale

Dans le contexte économique concurrentiel actuel, l'information joue un rôle important dans le quotidien des entreprises. L'acquisition, l'analyse et l'exploitation des informations sont devenues des choix stratégiques inévitables. La maitrise de l'information est une compétence capitale pour toute entreprise voulant s'imposer dans les premiers rangs de son domaine d'activité. A la lumière de ces impératifs, les grands volumes de données de production, relatifs à l'activité de l'entreprise, sont devenus de véritables mines de connaissances. A partir de ce moment, de gros efforts sont à déployer pour maitriser les grandes masses de données d'une part, et pour extraire des connaissances potentielles à partir de ces données. D'autre part les entrepôts de données (data warehouses) ont apporté une solution adéquate et efficace aux problèmes du stockage et de la gestion des données. Ainsi la technologie OLAP repose sur des outils pour la visualisation, la structuration et l'exploration des cubes de données. Il fournit des moyens aux utilisateurs pour naviguer dans les données multidimensionnelles afin d'y découvrir des informations pertinentes, mais ne permet pas de traiter les données complexes (texte, image…)

Notre problématique s'articule autour de l'analyse multidimensionnelle du contenu de documents, principalement constitué de données textuelles. L'environnement OLAP actuel ne prend pas en compte l'analyse de données textuelles l'absence de ces données peut mener un risque d'erreur pour la prise de décision.

D'un autre côté, la fouille de données (data mining) est une discipline qui a largement fait ses preuves depuis le début des années 90. Aujourd'hui, on peut considérer la fouille de données comme une nécessité imposée par le besoin des entreprises de valoriser les données qu'elles collectent dans leurs bases de données. La fouille de données emploie des méthodes d'apprentissage afin d'induire des modèles de connaissances exprimés dans des formalismes valides et compréhensibles.

Le but de notre travail est donc, de proposer un couplage entre l'analyse en ligne OLAP et la fouille de données, afin de définir un opérateur d'agrégation adapté à la nature des données textuelles.

Partie I

Etat de l'Art

I. Introduction :

L'informatique décisionnelle constitue un domaine évoluant en permanence. Depuis la définition du concept d'entrepôt de données par William H, le marché des solutions décisionnelles basées sur les entrepôts et outils OLAP (On Line Analytical Processing). Les secteurs de l'analyse en ligne OLAP et des analyses avancées de données du type fouille de données connaissent une croissance particulièrement marquée. L'informatique décisionnelle constitue une thématique majeure dans le monde de l'industrie et de la recherche.

Les systèmes décisionnels sont un ensemble de technologies destinées à permettre aux collaborateurs d'avoir accès et de comprendre les données de pilotage plus rapidement, de telle sorte qu'ils prennent des décisions meilleures et plus rapides pour atteindre les objectifs de leur organisation. Dans leur version la plus complète, ils permettent de répondre aux questions suivantes : Que s'est-il passé et pour quelles raisons ? Que va-t-il se passer ? Que vient-il de se passer ?

Un système décisionnel bien conçu doit donc :

- fournir un accès à des données fiables ;
- aider l'utilisateur à comprendre ces données. Le problème est moins aujourd'hui l'accès à l'information que la capacité à l'analyser, à la synthétiser
- faciliter la prise de décision : Connaître la signification d'une information, c'est bien. Savoir quoi en faire, c'est mieux.
- aider à la diffusion de l'information et à la mise en œuvre des actions.

II. Historique et définition :

La gestion de l'information réclame forcément, d'une façon ou d'une autre que celle-ci soit stockée sur un support. Au début de l'informatique moderne, les entreprises avaient choisi d'enregistrer l'information sous forme fichiers dit indexés. Les applications fonctionnaient alors de manière déconnectée. En effet, chacun des sous-systèmes gère une partie de l'information qui est, elle aussi, manipulée par un autre service. On avait donc une importante redondance d'information.

Depuis les débuts des années 1980 et avec une explosion au milieu des années 1990, les entreprises s'équipent de solutions de gestion et stockent un volume d'information qui s'agrandit un peu plus chaque jour. Rapidement apparait l'idée d'exploiter toutes ces données afin d'optimiser la décision. Naît ainsi l'informatique décisionnelle dont le but est de collecter, consolider, synthétiser l'information pour aider à la prise de décision. Partant de ce constat, le Docteur Edgar F. Cood, lui-même, publie, en 1993, un papier intitulé « Providing OLAP (On-line Analytical Processing) to UserAnalysts: An IT Mandate ».

Les bases de données OLAP sont des bases de données multidimensionnelles destinées à des analyses complexes sur des données.

Les systèmes OLAP doivent :

1. Supporter les exigences complexes des décideurs en termes d'analyse.
2. Analyser les données à partir de différentes perspectives (dimensions métiers).
3. Supporter les analyses complexes impliquant des ensembles de données volumineux.

III. Architecture:

Les outils décisionnels sont basés sur l'exploitation d'un système d'information décisionnel alimenté grâce à l'extraction de données diverses à partir des données de production et d'informations concernant l'entreprise.

Un outil appelé ETL (Extract, Transform and Load) est ainsi chargé d'extraire les données dans différentes sources, de les nettoyer et de les charger dans un entrepôt de données.

Enfin des outils d'analyse décisionnelle permettent de modéliser des représentations à base de requêtes afin de constituer des tableaux de bord, on parle ainsi de reporting.

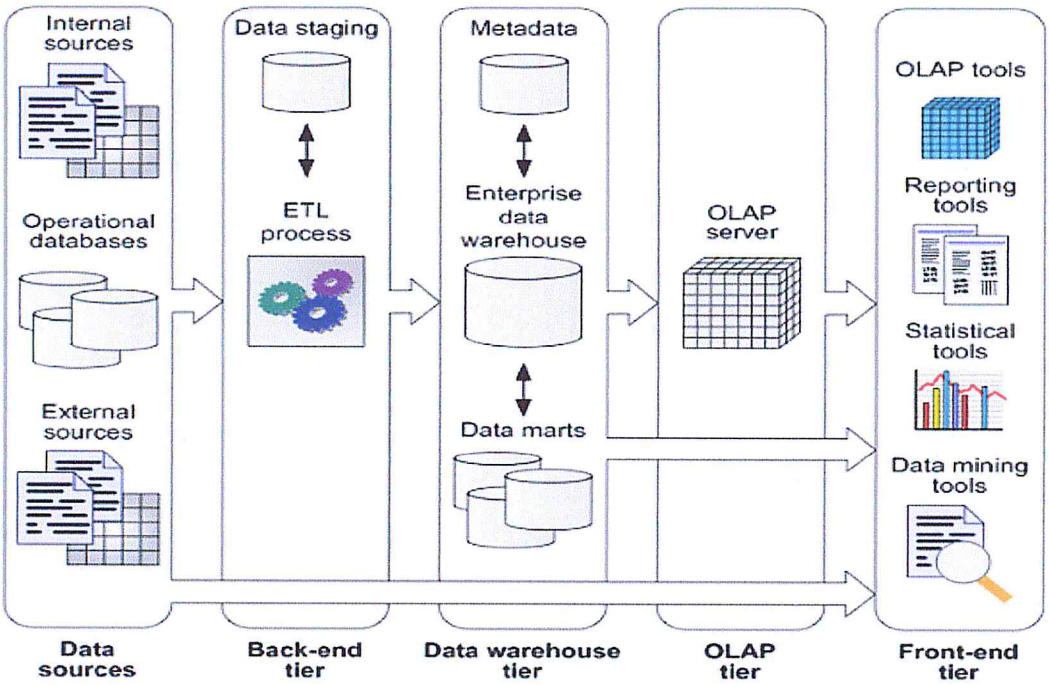


Figure 1 : *Schéma global d'un système décisionnel [Marlyse et Ghilani]*

IV. ***Processus détaillé :***

1. La phase de collecte :

La collecte s'effectué à partir de données appelées les données sources. Ces données peuvent se présenter sous différents formats. Il peut s'agir de fichiers "plats" (fichiers CSV avec séparateurs, fichiers XML, fichiers ASCII...) mais aussi de systèmes de bases de données (export de base MySQL, PostgreSQL, DB2, ORACLE...). Ces sources de données sont donc en général hétérogènes c'est pourquoi il va falloir passer par une phase dites d'intégration pour pouvoir les manipuler avant de les stocker dans notre système d'aide à la décision.

2. La phase d'intégration :

C'est à ce niveau qu'apparait la première couche logicielle de l'environnement décisionnel à savoir l'ETL. Cette couche offre des fonctions d'extraction de données issues de différents systèmes (internes ou externes), de transformation de ces données (homogénéisation, filtrage, calcul) et de leur chargement dans un **ODS** intermédiaire

ou directement dans le DW (entrepôt de données). Elle garantit la délocalisation de la charge de calcul et une meilleure disponibilité des sources.

La deuxième couche logicielle est l'ODS qui fait office de structure intermédiaire destinée à stocker les données issues des systèmes de production opérationnelle. Ce sont en quelque sorte des zones de préparation avant l'intégration des données dans le DW : périodicité journalière, données qualifiées, premier niveau de filtrage et d'agrégat. En général, il existe deux types de schéma : un schéma "ODS brut" qui contient les tables qui reçoivent les données brutes des différentes sources et un schéma "ODS final" qui contient des tables avec une structure (champs et contraintes associées) le plus proche possible du schéma du DW (même si les tables de celui-ci peuvent contenir plus de champs que les tables du DW) car ces données vont ensuite être figées dans l'entrepôt. L'ODS ne contient des données que sur une **faible période** et ces données vont être manipulées, transformées, traitées, modifiées plusieurs fois avant d'être copiées dans le DW. On peut se passer d'utilisation d'un ODS dans un seul cas : si les données du DW sont une simple copie (c'est-à-dire qu'il n'y a pas de traitements à faire et que les données extraites ne vont pas évoluer) des données de production (sources) ce qui n'est malheureusement pratiquement jamais le cas dans de grosses structures [1].

3. *La phase d'organisation*

La troisième phase permet de stocker les données dans un entrepôt appelé Datawarehouse. Le concept d'entrepôt de données a été proposé par W. H. Inmon en 1990 pour répondre à des besoins d'analyse pour les décideurs que les systèmes transactionnels ne pouvaient pas fournir. Ralph Kimball propose la définition suivante : Un entrepôt de données est un espace de stockage centralisé sur lequel repose un système décisionnel, son rôle est d'intégrer et de stocker l'information utile aux décideurs et de conserver l'historique des données pour supporter les analyses effectuées lors de la prise de décision.

Plus précisément, dans l'ouvrage "Le Data Warehouse" de J. M. Franco un entrepôt de données est une collection de données intégrées, thématiques, non volatiles et historiées pour la prise de décisions.

- **Des données intégrées** : Un entrepôt de données concerne les différents services et métiers de l'entreprise. Avant d'être intégrées dans l'entrepôt, et dans un souci de cohérence, les données doivent être mises en forme et unifiées. L'intégration nécessite une forte normalisation, une bonne gestion des référentiels et de la cohérence, une parfaite maîtrise de la sémantique et des règles de gestion s'appliquant aux données manipulées.

- **Des données thématiques** : Les données correspondent à des éléments d'analyse représentatifs des besoins utilisateurs. Elles constituent déjà un résultat d'analyse et une synthèse de l'information contenue dans le système décisionnel, et doivent être facilement accessibles et compréhensibles.

- **Des données non volatiles** : Afin de conserver la traçabilité des informations et des décisions prises, les informations stockées au sein de l'entrepôt de données ne peuvent pas être supprimées. Une requête lancée à différentes dates sur les mêmes données doit toujours retourner les mêmes résultats. Une donnée introduite dans l'entrepôt ne pourra donc plus être supprimée ni même modifiée. C'est pourquoi les données ne sont pas volatiles.

- **Des données historisées** : Chaque nouvelle insertion de données en provenance du système de production ne détruit pas les anciennes valeurs, mais crée une nouvelle occurrence de la donnée. L'historisation est nécessaire pour suivre dans le temps l'évolution des différentes valeurs des indicateurs à analyser. Ainsi, un référentiel temps doit être associé aux données afin de permettre l'identification dans la durée de valeurs précises. Pour gérer physiquement et sémantiquement l'ensemble des données, l'entrepôt de données doit nécessairement disposer de "données sur les données", à savoir de méta-données.

Une fois ces données stockées dans le Datawarehouse, on va pouvoir créer des magasins de données appelés : Datamarts. Comme le Datawarehouse, c'est un entrepôt de données mais dédié à une fonction de l'entreprise pour des raisons d'accessibilité, de facilité d'utilisation ou de performance.

Les données sont généralement équivalentes à celles présentes dans le DW principal mais elles sont représentées de façon adaptée aux besoins spécifiques de la fonction et/ou du domaine utilisateur (par exemple, on va créer un DM dédié pour le

service Marketing ou Commercial). Les magasins de données peuvent être perçus comme des petits entrepôts constitués d'un ensemble de données correspondant à un sujet précis, rendant très rapide les temps de réponses aux requêtes.

4. *La phase de restitution :*

La dernière phase concerne la restitution des résultats, on distingue à ce niveau plusieurs types d'outils différents :

- ✓ Les outils de reporting et de requêtes.
- ✓ Les outils d'analyse.
- ✓ La phase de Datamining.

Les outils de reporting et de requêtes permettent la mise à disposition de rapports périodiques, pré-formatés et paramétrables par les opérationnels. Ils offrent une couche d'abstraction orientée métier pour faciliter la création de rapports par les utilisateurs eux-mêmes en interrogeant le datawarehouse grâce à des analyses croisées. Ils permettent également la production de tableaux de bord avec des indicateurs de haut niveau pour les managers, synthétisant différents critères de performance.

Les outils **d'analyse** OLAP permettent de traiter des données et de les afficher sous forme de cubes multidimensionnels et de naviguer dans les différentes dimensions. Cet agencement des données permet d'obtenir immédiatement plusieurs représentations d'un même résultat, en une seule requête sous une approche descendante des niveaux agrégés vers les niveaux détaillés (Drill-down, Roll-up).

Les outils de **Datamining** offrent une analyse plus poussée des données historisées permettant de découvrir des connaissances cachées dans les données comme la détection de corrélations et de tendances, l'établissement de typologies et de segmentations ou encore des prévisions. Le Datamining est basé sur des algorithmes statistiques et mathématiques, et sur des hypothèses métier [1].

V. Conclusion :

Dans ce chapitre, nous avons donné un aperçu global sur les systèmes décisionnels car, avant de se lancer dans la panoplie d'outils qui font du décisionnel sur des données textuelles, sans doute faudrait-il d'abords connaître leurs mode de fonctionnement sur les données classiques (numériques).

Chapitre II

Etude bibliographique sur les travaux traitant l'agrégation de données texte

I. *Introduction* :

Les entrepôts de données ont été introduits pour répondre aux besoins grandissants des décideurs. Ceux-ci souhaitaient alors être munis de bases de données non pas dédiées au stockage robuste de leurs données pour répondre à des requêtes simples et répétitives (bases de données transactionnelles) mais plutôt à une représentation de leurs données en vue de prendre les meilleures décisions et répondre à des requêtes non répétitives et plus complexes. Le modèle multidimensionnel a alors été proposé pour répondre à ce besoin, et permet d'étudier un ensemble d'indicateurs (ou mesures) en fonction de plusieurs dimensions, chaque dimension pouvant être munie d'une ou plusieurs hiérarchies. Les opérateurs OLAP permettent de naviguer de manière intuitive dans de telles données multidimensionnelles (par exemple pour visualiser les données à différents niveaux de hiérarchies).

Quelques travaux récents se sont intéressés à intégrer les données textuelles dans un contexte d'entrepôt de données. Dans ce cadre, des méthodes d'agrégation adaptées aux données textuelles ont été proposées. Par exemple, les travaux de [Keith et al. 2005] proposent d'utiliser des approches de TALN (Traitement Automatique du Langage Naturel) pour agréger les mots ayant la même racine ou les mêmes lemmes (connaissances morphosyntaxiques). Les auteurs proposent également de rassembler les mots sur la base de classifications sémantiques généralistes existantes (WordNet et Roget). Outre l'utilisation de connaissances morphosyntaxiques et sémantiques pour agréger les données textuelles, d'autres travaux utilisent des approches numériques issues du domaine de la Recherche d'Information (RI) pour agréger les données textuelles. Ainsi, Lin agrège les documents sur la base des mots-clés présents dans ces derniers en utilisant une hiérarchie sémantique des mots présents dans l'entrepôt et des mesures issues de la RI. De telles méthodes issues de la RI sont aussi utilisées dans les travaux de Pérez-Martinez qui consistent à prendre en compte une dimension "contexte" et "pertinence" pour construire un entrepôt de données textuelles appelé R-Cube. Certaines approches proposent d'ajouter une nouvelle dimension spécifique. Par exemple : une dimension **topic** et appliquent l'approche PLSA pour extraire les thèmes représentatifs des documents dans cette nouvelle dimension. Enfin, Pujolle et leurs membres proposent d'agréger des parties de documents afin d'offrir au décideur des mots-clés caractéristiques propres à cette agrégation. Dans ce cadre, les auteurs

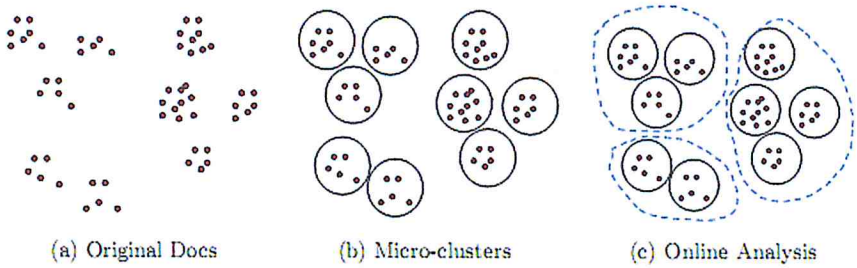


Figure 2: Exemple illustratif de micro-clusters [Zhang et al, 2009]

Après la réalisation de MiTexCube, on peut faire l'analyse en ligne basées sur cette nouvelle représentation. Noter qu'il existe trois défirent tâches d'analyse en ligne qui sont :

- **Résumé standard d'une cellule** : Permet de générer un résumé d'une cellule en **p** différents groupes (micro-cluster), tel que le nombre **p** est spécifie par l'analyste à travers l'algorithme K-means.
- **Résumé d'une cellule par requête** : Le but dans cette consiste à personnaliser un résumé basé sur le thème qui est préféré par l'analyste.
- ***Comparaison sujet commun.*** Une autre tâche de l'analyse consiste à comparer les multiples cellules de texte pour révéler la différence de leur couverture sur des thèmes communs. [Zhang et al, 2009].

III. Fonctions d'agrégation :

Les fonctions d'agrégation sont un élément important de la génération de rapports sur des bases de données. La spécification de fonctions d'agrégation dans les bases de données relationnelles a été une problématique active depuis la définition de l'algèbre et du calcul relationnel défini par codd. Pour une application au sein des bases de données statistiques en se reposant sur la notion de «summary tables », tables bidimensionnelles à l'origine des tableaux croisés dynamiques.

Dans ce qui suit, nous allons dans un premier temps donner un aperçu sur les fonctions d'agrégation classiques, par la suite nous allons exposer quelques travaux de recherche qui proposent des fonctions d'agrégation avancées.

1. Fonctions d'agrégation classiques

Les entrepôts de données classiques sont accompagnés d'une série de fonctions d'agrégation classiques .Il s'agit de fonctions simples qui regroupent un ensemble de valeurs en une valeur unique. Parmi ces fonctions on trouve généralement les cinq fonctions suivantes :

Somme (SUM) : cette fonction retourne la somme numérique de l'agrégat

Comptage (COUNT) : cette fonction compte le nombre d'instances dans un agrégat

Minimum (MIN) : cette fonction retourne la plus petite valeur d'un agrégat

Maximum (MAX) : cette fonction retourne la plus grande valeur d'un agrégat

Moyenne (AVERAGE) : cette fonction retourne la valeur moyenne d'un agrégat.

Ce jeu de fonctions a été augmenté de fonctions statistiques pour permettre la génération de rapports plus complets. Il s'agit essentiellement de fonctions statistiques, telles que le calcul de moyennes mobiles, de barycentres ou encore de médianes ou d'écart type. De plus les SGBD récents (par exemple Oracle) offrent une interface de programmation permettant à un programmeur de spécifier ses propres fonctions d'agrégation.

2. Fonctions d'agrégation avancées

Dans les dernières années il y a plusieurs fonctions d'agrégation sont évoluées pour analyser les déférents type de données, Ces fonctions proviennent de différents domaines :

- ✓ Le multidimensionnel
- ✓ Les systèmes d'information géographiques (SIG)
- ✓ La fouille de données.

Le décisionnel a vu la création d'un opérateur de regroupement CUBE, qui emploie de manière intensive les fonctions d'agrégation dans un environnement décisionnel. Il s'agit d'un opérateur calculant les totaux généralisés d'une sélection de données.

Dans le cadre des SIG, un domaine décisionnel apparu, notamment avec le SOLAP (Spatial: OLAP). Des fonctions spécifiques adaptées aux données géographiques virent le jour. Les données géographiques étant stockées sous les formes de points, segments et surfaces, les fonctions adaptées se chargent de permettre un regroupement de ces types de données (avec par exemple le barycentre de plusieurs points, la surface moyenne,…).

Récemment des fonctions d'agrégation issues de la fouille de données commencent à voir le jour au sein de l'environnement OLAP, parmi ces fonctions *Skline* Il s'agit d'une fonction cherchant à résoudre le problème de maximisation de vecteurs (Vector Maximisation Problem—VMP) et de rechercher une solution maximale ou minimale pour un problème à (au moins) deux variables. Par exemple cette fonction permet de rechercher les hôtels les moins chers en fonction de leur distance à la plage voisine. Dans ce cas précis, il s'agit de trouver les hôtels qui minimisent le coût et la distance.

Dans la même lignée, des opérateurs de classification issus de fouille de données ont été inclus dans les systèmes OLAP. Parmi ceux-ci, OpAC [Messaoud et al ,2004] est un opérateur regroupant les instances selon une classification ascendante hiérarchique (CAH). Néanmoins, ces fonctions ne sont guère utiles dans l'environnement OLAP car elles détruisent systématiquement l'agencement hiérarchique des axes d'analyses, réduisant à néant les possibilités de forage par la

suite. Ainsi les résultats obtenus par ce type de fonction d'agrégation sortent de l'environnement OLAP car de nombreuses opérations de manipulation ne peuvent plus s'y appliquer.

2.1 Les opérateurs d'agrégation basés sur la fouille de données :

Parmi les travaux de recherche proposant un couplage entre l'analyse OLAP et la fouille de données, nous retrouvons celui de [Ben-messaoud et al ,2004]. Ce dernier défini deux opérateurs Olap qui sont : ORCA et OPAC. Le premier (ORCA) est un opérateur de réarrangement d'un cube par analyse factorielle (ACM) et le second (OPAC) est un opérateur d'agrégation dans un cube de données par une CAH (Classification Ascendante Hiérarchique).

Dans la suite de ses travaux [Ben-messaoud, 2006] rajoute un nouvel opérateur appelé AROX, dédie à l'explication dans un cube de données. Ce dernier se base sur l'approche d'extraction des règles d'association à partir des cubes de données.

2.1.1 *OPAC :*

OpAC (Opérateur d'Agrégation par Classification) consiste particulièrement en l'agrégation sémantique des modalités d'une dimension d'un cube de données en se basant sur la technique de la classification ascendante hiérarchique.

Dans [Ben-Messaoud et al ,2004], ils utilisent la classification ascendante hiérarchique (CAH) en vue de construire des classes correspondant à de nouveaux agrégats dans le cube. Ainsi, la classification est perçue comme une technique d'agrégation sémantique dans les cubes de données. Dans cette approche, la mise en œuvre de la classification dans les données multidimensionnelles se base sur la structuration et la classification de couplage entre l'analyse en ligne et la fouille de données. des opérations OLAP sont utilisés afin d'extraire les données, notamment les individus et les variables, nécessaires à la classification. Dans [Ben-Messaoud et al 2, 2004], ils ont introduit une première formalisation de cette approche de classification dans les cubes de données. Dans [Ben-Messaoud et al, 2006], ils ont amélioré et appliqué l'approche à un cas de données complexes. Ce cas d'application concerne des données de mammographies relatives à des dossiers de patientes atteintes du cancer

du sein. L'idée de base de l'opérateur OpAC consiste à exploiter les mesures contenues dans un cube de données afin d'agréger les modalités d'une de ses dimensions.

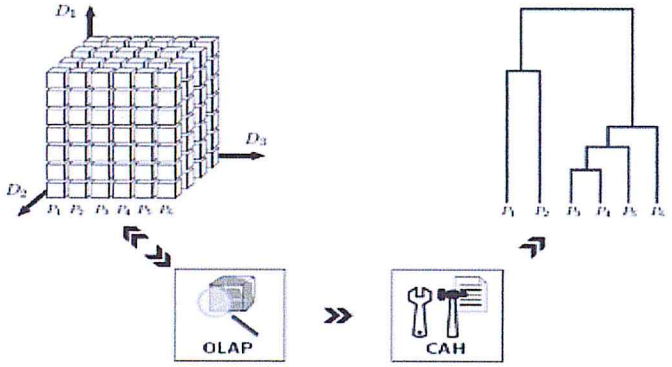


Figure 3 : Etapes de l'agrégation par classification dans les cubes de données [Ben-Messaoud et al ,2004]

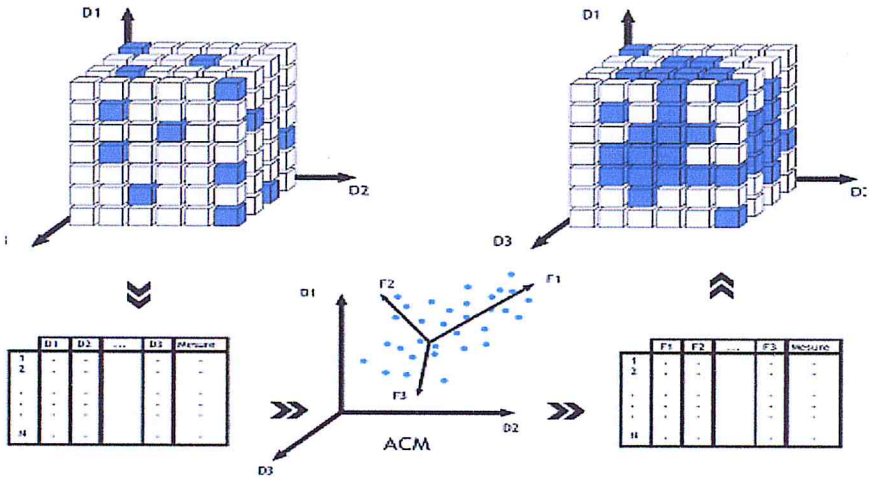
2.1.2 ORCA :

Les opérateurs OLAP classiques permettent de: naviguer, explorer et résumer un cube et détecter des régions intéressantes dans le cube. Mais, dans des cubes épars et de grande taille il aura une navigation et exploration difficile ainsi qu'un manque d'outils automatiques. En outres, les modalités de dimensions sont représentées selon un ordonnancement lexical préétabli qui correspond souvent à un ordre naturel (ordre chronologique pour les dates et alphabétique pour les libellés par exemple.) Par conséquent, les points associés aux faits observés (les cellules pleines) sont éparpillés dans l'espace des dimensions d'un cube de données. Pour améliorer la visualisation des données dans les cubes, ils ont proposé une méthode qui consiste à coupler l'analyse en ligne avec l'analyse des correspondances multiples (ACM) [Benzécri, 1973] . Cette proposition se base sur la transformation des données multidimensionnelles en données tabulaire afin de les exploiter par des algorithmes de fouille de données.

La 1ére étape consiste à transformer les données initiales d'un cube en tableau individus-variables selon un codage binaire spécifique à l'ACM. Dans la 2éme étape, ils appliquent l'ACM aux données transformées afin d'obtenir des axes factoriel représentant aux mieux les faits OLAP et traduisant des relations avec les modalités des dimensions du cube, chaque axe factoriel (ou facteur) est caractérisé par une

valeur propre indiquant l'inertie (dispersion) des individus dans la direction définie par l'axe. D'où l'intérêt d'une méthode de réorganisation des données multidimensionnelles pour réduire l'effet de leur éparsité, dans cette méthode, ils utilisent l'ACM comme étant un outil d'aide à la construction de cubes de données ayant de meilleures caractéristiques pour la visualisation.

Figure 4 : étapes de la réorganisation d'un cube de données par approche factorielle



[Ben-Messaoud et al ,2004]

2.1.3 *AROX :*

Différemment aux deux premiers opérateurs, cette méthode adapte un algorithme de fouille afin d'extraire des connaissances directement à partir de la structure multidimensionnelle des données. Cette proposition s'inscrit dans une démarche explicative dans les cubes de données en se basant sur les règles d'association. Dans [Ben-messaoud, 2006], les auteurs mettent en place un nouvel algorithme, de type Apriori, pour une recherche guidée par des règles d'association dans les cubes de données. Une visualisation graphique des règles d'association extraites est également proposée afin de mieux valoriser les connaissances qu'elles véhiculent. La technologie OLAP se limite à des tâches exploratoires et ne fournit pas d'outils automatiques pour expliquer les relations et les associations potentiellement existantes entre les données d'un cube.

Beaucoup d'études ont abordé le problème de l'extraction des règles d'association à partir des cubes de données. Cette proposition de couplage entre l'analyse en ligne et la fouille de données se base sur une approche qui adapte plutôt l'algorithme de la fouille aux données multidimensionnelles. Ainsi, ils introduisent un nouvel algorithme pour la recherche des règles d'association directement à partir des cubes de données sans transformation préalable de ce dernier. Dans le cadre général pour la recherche de règles d'association à partir des cubes de données. Ils utilisent le concept des métarègles inter dimensionnelles afin d'offrir à l'utilisateur la possibilité de guider le processus de fouille vers des contextes d'analyse ciblés qui répondent à ses besoins d'explication et à partir desquels seront extraites les règles d'association.

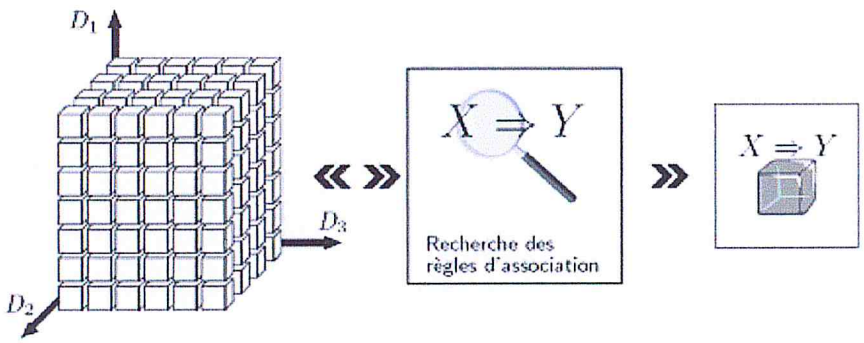


Figure 5 : Etapes de l'explication dans les cubes de données par règle d'association [Ben-messaoud, 2006].

2.1.4 *RoK* :

Les auteurs ont proposé un nouvel opérateur d'agrégation **RoK** (Rol l-up with K-means) en utilisant la fouille de données. L'opérateur RoK permet de créer un nouveau niveau de granularité dans une hiérarchie de dimension en se basant sur une méthode de classification automatique. L'opérateur RoK utilise la méthode des k_means qui permet de rechercher des structures naturelles dans les données. Il s'agit dans ce cas, de trouver un bon regroupement des instances d'un niveau d'analyse existant choisi par l'utilisateur, à partir duquel un nouveau niveau d'analyse peut être créé. Cette approche enrichit considérablement l'analyse multidimensionnelle car elle

offre de nouveaux angles de vues intéressants sur les faits pouvant être explorés par l'utilisateur [Bentayeb et Favre 2009].

2.2 Les fonctions d'agrégation basés sur la fouille de texte :

Quelques travaux de recherche suggèrent l'emploi de techniques de fouille de texte pour l'analyse du contenu de documents. Plusieurs fonctions sont proposées :

- **SUMMARY** : permet la génération d'un résumé du texte agrégé
- **TOP_KEYWORDS** : sélectionne les n principaux mots clefs du texte à agréger
- **TOPIC** : extrait le sujet d'un bloc de texte.
- **CLUSTERING** : est une fonction qui partitionne des textes en fonction de leur contenu.

Dans la suite, nous allons détailler les deux fonctions d'agrégation : Top-kw, AVG-K. Ces dernières sont des fonctions opérant sur des mots clef. Ces mots clef peuvent avoir été extraits du texte lors de l'alimentation des documents dans le magasin de données ou bien ils peuvent être extraits à la volée lors de l'analyse par une fonction d'agrégation opérant sur des fractions de texte. Proposée par [Ravat et al 2007]

- **TOP_KW$_k$** : permet l'agrégation d'un ensemble de documents en ses k termes les plus représentatifs, qui exploite la fonction de pondération TF.IDF issues de la recherche d'information
- **AVG_KW** : permet de résumer un ensemble de mots-clefs issus d'un vocabulaire contrôlé par un ensemble limité de termes plus généraux. Cette fonction repose sur une ontologie légère de domaine.

2.2.1 Top_KW$_k$:

La fonction d'agrégation TOP_KWk extrait les k termes les plus représentatifs d'une mesure textuelle brute constituée de n termes (ou mots). Afin de déterminer les k termes les plus représentatifs, ils ont adapté au contexte OLAP des techniques bien maîtrisées en Recherche d'Information (RI) qui ordonnancent les termes selon leur représentativité en fonction de poids. Pour ce faire, il est nécessaire

de connaître la représentativité d'un terme vis-à-vis de la collection (intégralité des autres documents). Dans le contexte OLAP, il n'est pas nécessaire de connaître cette représentativité vis-à-vis de la collection complète, mais plutôt selon les documents qui seront agrégés par la fonction. Le problème est alors d'opérer sur une liste variable de documents qui change à chaque manipulation multidimensionnelle.

2.2.2 AVG_KW :

La fonction d'agrégation AVG_KW est conçue pour synthétiser un ensemble de mots clef issus d'un vocabulaire contrôlé en un ensemble plus petit de mots clef plus généraux. La fonction prend en entrée un ensemble de mots clef, chacun associé à une distance et génère un nouvel ensemble de mots clef agrégés. Le processus d'agrégation se base sur l'ontologie de domaine qui est une ontologie légère ou encore une ontologie dotée de liaisons «est : un informelles». Ce type d'ontologie correspond à unehiérarchie de concepts d'un domaine où chaque nœud représente un concept (un mot : clef) et chaque lien entre les nœuds modélise une relation plus complexe que larelation «est : un».

Les mots clef sont tous issus du vocabulaire contrôlé représenté par l'ontologie et le domaine de l'ontologie est proche du domaine des documents à analyser. Pour chaque paire de mots clef, la fonction trouve le plus petit ancêtre commun correspondant. Mais lors de l'agrégation de mots clef très éloignés dans l'ontologie, il y a une très forte probabilité de retourner systématiquement le mot clef représenté par le nœud racine de l'ontologie. Afin d'éviter ce phénomène, une limite dans le processus d'agrégation doit être imposé. En effet, plus les mots clef sont éloignés les uns des autres,plus l'agrégation ne se traduit par une perte de sens. Pour surmonter ce problème, la fonction emploie une distance maximale autorisée lors de l'agrégation de mots clef : DMAX. Pour l'instant, des heuristiques informelles suggèrent une distance comprise entre 3 et 5. Il est à noter qu'avec une ontologie généraliste telle que WordNet « Ontologie lexicale anglaise » DMAX est plus de l'ordre de 3. A ce jour et à notre connaissance,dans la recherche concernant les ontologies, ce problème n'a pas encore été résolu.

IV. Conclusion :

Dans cette partie, nous avons exposé les différentes solutions aux problèmes de l'analyse des données complexes, particulièrement ceux de l'analyse en ligne des données textuelles. Nous avons constaté que le couplage entre l'analyse en ligne et la fouille de données est capable d'enrichir et de rehausser le processus décisionnel. De plus, la fouille de données a déjà fait ses preuves pour l'extraction des connaissances à partir des données textuelles. Par conséquent, la fouille de données est capable d'étendre les capacités de l'OLAP pour analyser ces données.

Chapitre III

Etude sur les méthodes de fouille de texte

I. Introduction

La fouille automatique de textes concerne les processus interactifs et itératifs de découverte de connaissances dans de grandes collections de documents. La fouille de textes est définie par ***Sebastiani*** comme l'ensemble des tâches qui permet l'analyse d'une grande quantité de textes et la détection de modèles fréquents, dans le but d'extraire l'information probablement utile. Fouille de données et fouille de textes possèdent en commun des méthodes et algorithmes tels que les algorithmes de recherche par niveaux de motifs vérifiant certaines propriétés, l'exemple le plus classique de propriété étant la fréquence.

La fouille de textes présente ses propres spécificités : les documents peuvent être plus ou moins structurés, la phase de prétraitement joue un grand rôle. Il est courant et souvent intéressant de prendre en compte l'ordre des mots, et il est maintenant établi que des ressources provenant du traitement automatique des langues et/ou de la linguistique sont nécessaires pour apporter des résultats applicatifs avec une réelle plus-value. Beaucoup de méthodes de fouille de textes incluent des traitements d'analyse pré-lexicale (exemple : traitement des chiffres), d'analyse lexicale telle que l'élimination de « mots vides », morphologie, analyse syntaxique (exemple : détermination des groupes nominaux), analyse sémantique incluant des apports linguistiques aussi bien que des particularités des textes étudiés.

Du côté de l'analyse linguistique, l'approche du texte par des marqueurs de surface est connue chez les linguistes de corpus. Elle est exploitée entre autres par **Péry-Woodley** qui a travaillé sur un corpus académique anglais langue maternelle ou langue seconde. Ses travaux portent également sur la relation entre marques linguistiques et mise en forme matérielle. Cependant, dans les analyses de discours contemporaines, et spécialement en informatique linguistique, l'analyse est implicitement limitée au paragraphe. L'approche liant stylistique et extraction d'information automatique a été expérimentée sur un corpus internet, principalement journalistique, étudié au niveau de la phrase. Les travaux de Hearst portent sur des groupes de paragraphes. La structure HTML du document est explicitement exploitée pour la recherche d'informations orientée Internet ou pour la génération de résumé mais non la structure linguistique.

II. le processus de fouille de texte :

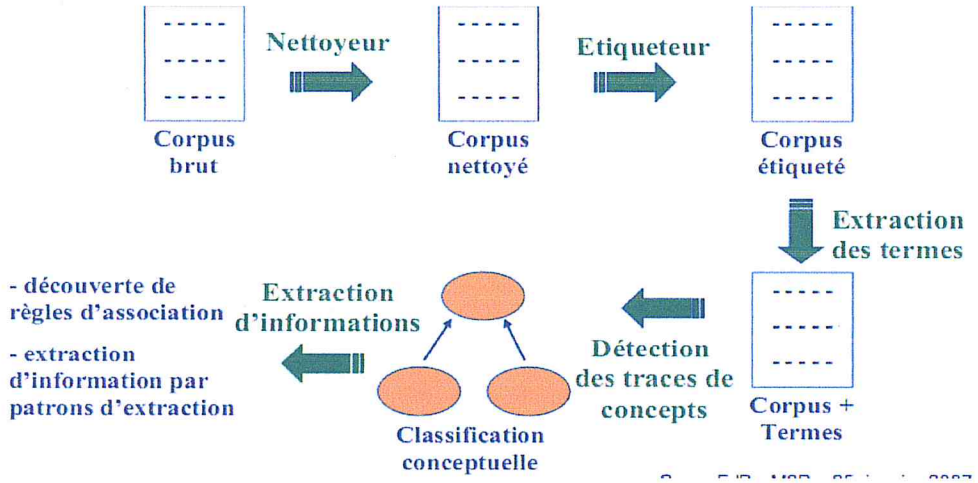


Figure 6 : processus de fouille de texte [Violaine et Yves, 2005]

1. Nettoyage :

Tout d'abord, un tri est généralement effectué pour ne prendre en compte que les mots significatifs dans les textes : c'est pour cela que les listes de "**mots vides**" ou "**anti dictionnaires**" peuvent être utilisés par exemple : « le », « la », « de », « du », « ce », « ça ».cette phase consiste à éliminer les mots vides de sens comme les articles et les prépositions avec une suppression de ponctuation et des caractères spéciaux

2. Etiqueteur :

Problèmes avec la notion de "mot" :

Qu'est-ce qu'un mot ? La première définition possible fait appel à un critère formel : un mot c'est ce qui, dans un texte, est *compris entre deux séparateurs*. Mais cette définition n'est pas aussi opératoire qu'on pourrait l'espérer. Qu'on songe aux cas suivants :

- **l'apostrophe :** est la plupart du temps la marque d'une séparation entre deux mots : "j'ai", "l'arbre", "d'un", etc. sont bien constitués de deux "mots". Pourtant, "aujourd'hui" et "prud'homme", malgré l'apostrophe qu'ils contiennent, ne sont généralement considérés que comme un seul mot.

- **le point** : est apparemment un séparateur assez puissant pour isoler les phrases. Mais suffit-il pour autant à empêcher que "I.B.M." ou "11.09.2001" ne semblent constituer qu'une seule unité ?
- **le tiret :** est un séparateur encore plus problématique : qu'est-ce qui nous autorise à ne trouver qu'un seul mot dans "porte-monnaie" ou" entre-temps" et à en trouver plusieurs dans "cet homme-là", "est-ce-que", "voulez-vous" ? Et combien de mots y a-t-il dans "y a-t-il" ou dans "c'est-à-dire" ?

même le **caractère blanc** n'est pas un séparateur fiable : on a bien envie de faire de "parce que" ou de "pomme de terre" un seul mot, tandis que "du" et "au", qui résultent d'un amalgamme ("du" pour "de le", "au" pour "à le"), en font plutôt deux à eux tous seuls.

3. ***Extraction des termes:***

Dans cette étape il y a plusieurs méthodes qui sont proposé pour l'extraction des termes parmi ces méthodes :

Le **TreeTagger** de [Schmid, 1994] estime la probabilité qu'un mot ait une étiquette grammaticale (Nom, Adjectif, Déterminant, etc) en s'appuyant sur des arbres de décision binaires [Quinlan, 1986]. Ces derniers sont construits récursivement à partir d'un ensemble de trigrammes connus (suites de trois étiquettes grammaticales consécutives constituant l'ensemble d'apprentissage).

La **racinisation** ou **stemming** en englais [Porter, 1980] les techniques utilisées pour ce faire reposent généralement sur une liste d'affixes (suffixes, préfixes, postfixe, antéfixes) de la langue considérée et sur un ensemble de règles de racinisation/désuffixation construites a priori qui permettent, étant donné un mot de trouver sa racine. La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son préfixe et son suffixe, à savoir son radical. Contrairement au **lemme** qui correspond à un mot réel de la langue, la racine ne correspond généralement pas à un mot réel. Par exemple, le mot « chercher » a pour radical « cherch » qui ne correspond pas à un mot réel. Par contre dans l'exemple de « frontal », le radical est « front » qui lui l'est.

L'étiqueteur de **Brill** appose une étiquette grammaticale à chacun des mots d'un texte en utilisant un lexique, des règles lexicales et des règles contextuelles. Dans l'approche développée dans les travaux de [Brill, 1994], l'auteur s'appuie sur un corpus d'apprentissage du **Wall Street Journal**. Le but est alors d'apprendre des règles d'étiquetage à partir de ce corpus. Ce corpus est annoté manuellement et représente l'ensemble des étiquetages corrects.

À chaque étape d'apprentissage, des règles sont modifiées et le résultat de l'étiquetage avec ces nouvelles règles est comparé avec le corpus représentant l'ensemble des annotations justes. Tant qu'un nombre d'erreurs seuil dans l'étiquetage subsiste, le processus d'apprentissage continue. Les transformations des étiquettes s'effectuent (1) en changeant une étiquette par une autre suivant les mots ou les étiquettes des mots proches, (2) en utilisant certaines caractéristiques pour les mots inconnus (lettres en majuscules pour les noms propres, suffixe des mots, etc).

4. Détection des traces de concepts :

Une fois le texte transformé en une représentation mathématique, on peut le classer parmi un ensemble de d'autres textes. Plusieurs techniques existent comme : Les centres mobiles, les nuées dynamiques et les K-Means.

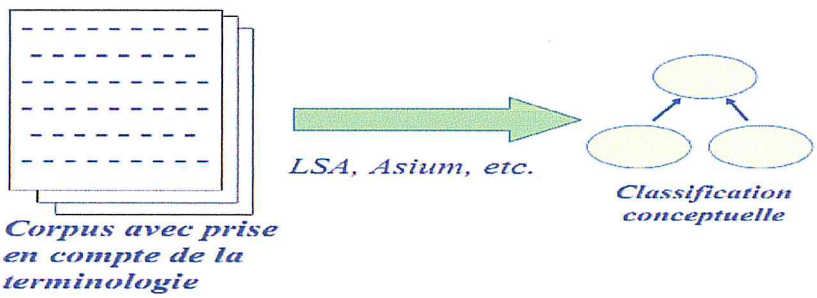


Figure 7: les méthodes des *détections des traces de concept* [Violaine et Yves, 2005]

5. Extraction d'informations:

Dans cette étape on utilise l'extraction d'informations par patrons d'extraction et par les règles d'association pour découvrir l'information pertinente.

III. Les différentes tâches de fouilles de texte :

Quand on parle de fouille de textes, on utilise un terme générique, traduction approximative de l'anglais «**text mining** », et l'interprétation la plus immédiate pourrait se référer à la recherche d'information, ou à l'extraction de connaissances. C'est en effet dans ces thématiques que la fouille de texte a pris naissance. Cependant, avec le succès grandissant de ce domaine scientifique, et surtout ses possibilités d'applications concrètes relativement spectaculaires, d'autres thématiques sont venues compléter ce premier ensemble.

1. La recherche d'information classique :

La recherche d'information RI traite des méthodes susceptibles de retrouver des textes ou des extraits de textes pertinents par rapport à une requête, cette dernière étant exprimée à l'aide du même matériau que ces textes, c'est-à-dire avec des mots. Cette tâche est en effet la plus classique, et la plus ancienne de la fouille de textes. Les principaux ateliers de TREC lui sont consacrés. La tâche de RI peux être mieux réalisée dans deux cas :

- **premièrement** quand on cherche une relation **sémantique (calculable)** entre les mots du texte [Yang et Chute 1992]
- **deuxièmement**, lorsque l'on adjoint au moteur des ontologies, ou des structures de concepts permettant de capturer des textes pertinents. Cela, grâce à des relations **lexicales** comme la **synonymie ou l'hyperonymie**, mais également grâce à des relations sémantiques d'appartenance, de sous-catégorisation, de participation ou d'attribution, relations présentes dans des réseaux de concepts, ou dans des arborescences de connaissances. Des réseaux tels que Word Net mélangent à la fois les relations lexicales et sémantiques. Ils permettent de marquer les mots aussi bien comme matériau (linguistique) que comme idées. Cette double étiquette a des avantages et des inconvénients. La polysémie (multiplicité des sens) est parfois difficile à suivre, et peut provoquer une surcharge dans la recherche et abaisser la précision des systèmes.

Les méthodes utilisées en RI :

La donnée fournie par l'utilisateur est donc une requête. Celle-ci peut prendre des formes diverses, suivant le niveau d'expertise de cet utilisateur et la structure de la base de documents à interroger : simple liste de mots clés, langage de requête structuré (combinaisons de critères booléens, expressions rationnelles, requêtes type SQL...), voire document « exemple » dont on cherche des exemplaires « proches » parmi un ensemble de textes.

Les ressources sollicitées sont tout d'abord le corpus de textes ou de documents que l'on cherche à interroger. Ce peut être une base d'articles, une encyclopédie, ce peut être Internet, il est éventuellement fait appel aux ressources nécessaires à la représentation de la requête par un vecteur.

Enfin, quand la requête est réduite à un ensemble de mots-clés, il est courant d'utiliser un thesaurus ou une ontologie pour l'étendre à des mots « proches » (par synonymie, ou par généralisation en « remontant » dans la hiérarchie par exemple).

On distingue trois familles de méthodes pour aborder la RI :

- les méthodes booléennes fonctionnent à l'aide d'un simple index qui donne, pour chaque unité lexicale figurant dans la requête, la liste des textes où cette unité est présente. Les requêtes acceptées sont alors généralement des combinaisons de critères booléens (avec les opérateurs NON, ET, OU). Des calculs simples permettent d'obtenir la liste des textes où tous ces critères sont satisfaits en même temps.

Les méthodes vectorielles, comme leur nom l'indique, codent toutes les informations (la requête et les documents de la base) sous la forme de vecteurs. La représentation TF-IDF est née dans ce contexte, et y est particulièrement efficace. La RI se ramené alors a trouver les vecteurs les plus « proches » d'une vecteur donné (celui représentant la requête). Pour quantifier ces distances, on utilise souvent des mesures basées sur le cosinus de l'angle qu'ils font entre eux (facile à calculer par des formules mathématiques).

- Les méthodes l'apprentissage automatique qui en fait reviennent à faire de la classification automatique en supposant que l'on connaît déjà, pour la requête, un ensemble de documents "pertinents" et de documents "non pertinents", et que l'on cherche a trouver tous les documents devant être classés comme pertinents. On le voit, cette méthode n'est pas vraiment comparables aux autres, puis qu'elle fait des hypothèses supplémentaires sur ce qui doit être fourni au système. Mais c'est la seule manière de faire intervenir de l'apprentissage automatique dans la tâche de recherche d'information.

2. ***L'extraction de connaissances :***

Les textes étant par définition des réservoirs de connaissances, ces derniers ont jusqu'à il y a environ quinze ans, été assez peu exploités tels quels. La médiation de l'esprit humain a toujours été requise pour la construction de ces réseaux, arbres ou ontologies, nécessaires aux systèmes à base de connaissances, et plus pragmatiquement, aux moteurs d'inférence un tant soit peu évolués. **Si les règles de production**, qui sont des modes d'emploi de la mise en relation des différents savoirs, devaient a priori provenir majoritairement de l'expertise des individus, les connaissances factuelles et classificatoires pouvaient être automatisées. C'est pourquoi, des branches thématiques issues d'une part de la représentation des connaissances, et d'autre part des bases de données, se sont penchées sur le texte comme source de connaissances modélisables, et les années 2000 ont vu l'explosion de l'acquisition des ontologies à partir de textes .

Pour ces deux communautés, cette source est dite non structurée, ou semi-structurée. En effet, il n'y pas eu d'intervention complémentaire de la part de l'Homme pour rajouter des éléments forçant les classifications. Néanmoins, pour les spécialistes de langage naturel, un texte est forcément une source de données structurées, puisqu'il est composé de phrases, de paragraphes et de sections éventuellement. Les phrases sont par définition des constructions élaborées visant à la fois à exprimer des idées mises en contexte, et à restreindre les possibilités sémantiques des mots en fonction d'un fil conducteur guidant la mise en œuvre du texte.

Un texte en langage naturel, pour un lecteur humain, est un ensemble parfaitement construit : ponctuation, contextualisation sont des éléments de forme et de fond déterminants pour la compréhension. Or pour que cette compréhension puisse être automatisée, il aurait fallu disposer d'algorithmes et de logiciels capables de détecter ses

structures à l'instar du lecteur. Cela signifie disposer d'un analyseur non seulement morphologique (lemmatiseur ou 'tagger') capable d'affecter les catégories grammaticales, mais également d'un véritable analyseur syntaxique, susceptible d'isoler les constituants et de détecter les dépendances. Constituants et surtout dépendances sont les véritables clés de la structuration de la phrase, indiquant au lecteur quels sont les éléments moteurs (agents, gouverneurs) et quels sont ceux qui sont compléments et dans quelle mesure ces derniers sont modifieurs. Certes cette structuration semble autre que celle requise pour la constitution de taxonomies ou d'ontologies, où les relations entre concepts sont les éléments à détecter de façon primordiale, mais elle existe, et surtout, il semble difficile de l'écarter, justement lorsqu'il faut relier les dits concepts. Voici quelques remarques de fond liées à ce problème.

- Considérer un texte comme **un sac de mots** revient à perdre de vue la mise en perspective de l'importance relative des éléments langagiers : le langage naturel est non commutatif dans ses relations de dépendance, il est relativement ordonné, et de nombreuses dépendances de type "complément" peuvent être des indices forts de relation d'attribution (attributs de concepts). Les exemples auxquels sont sensibles actuellement les chercheurs appartiennent à l'ordonnancement des groupes nominaux prépositionnels ou adjectivaux. Ainsi "voile de bateau" est une sous-classe du concept "voile", alors que "bateau à voile" est une sous-classe du concept "bateau". De la même manière, "moyenne pondérée" et "poids moyen" faisant l'un et l'autre référence aux concepts de "moyenne" et de "poids" mettent une relation d'attribution différente dans les deux cas. Dans le premier exemple, c'est le poids qui est un attribut de la moyenne, pour générer éventuellement une sous-classe de ce concept, et dans le second, la moyenne est un attribut permettant de spécifier un type de poids particulier.

La perte d'information en transformant le texte en ensembles non ordonnés de mots ou de termes, même complexe, est donc, pour cette tâche, relativement importante. Les quelques résultats actuellement obtenus sont plus satisfaisants pour les esprits susceptibles de compléter les manques observés, par leur propre action, que si on devait considérer ces résultats de la manière la plus détachée et la plus objective. D'ailleurs, à ce sujet, certains auteurs se posent la question de la valeur des objets extraits par rapport à leur problématique d'une part, et par rapport aux exigences

réelles de la tâche d'autre part. Extraire des connaissances pour enrichir une base existante, ou pour vérifier les éléments de cette dernière suppose de récupérer véritablement l'ensemble des relations,

car les dépendances ont un rôle sémantique et font l'articulation entre syntaxe (construction) et sémantique (interprétation). Les travaux qui ont été réalisés jusqu'à présent dans ce domaine se sont fondés sur les particularités des domaines dont il fallait extraire des connaissances. Plutôt que des cadres généraux, les recherches se sont focalisées sur des domaines techniques ou scientifiques relativement limités, à vocabulaire restreint. Les connaissances majoritairement extraites sont factuelles et classificatoires, et donc peuvent se confondre avec la terminologie.

Une première observation permet d'associer la structuration linguistique en groupes nominaux et/ou prépositionnels avec les concepts du domaine. Des extractions de ces groupes permettent de baliser l'univers conceptuel de l'ontologie à construire ou à compléter. Quant à la découverte des liens entre concepts, pour l'instant, c'est le repérage des règles d'association qui semble primer. Si embolie pulmonaire est liée avec phlébite ou thrombose dans le corpus, une association entre ces concepts est détectable à l'aide de diverses techniques. Mais la nature précise de l'association, portée par le prédicat ou par la sémantique de la phrase (et non pas seulement des mots) échappe encore aux extracteurs de connaissance qui font l'impasse sur la compréhension syntaxique et sémantique du texte en langage naturel.

3. *La classification de documents :*

La classification de documents s'est imposée à partir de la remise au goût du jour de la classification de données (numériques, génomiques, etc.). Issue majoritairement de la statistique, la méthodologie scientifique vise essentiellement à récupérer des groupes relativement homogénéisés auxquels on attribuera le vocable de classe (si ces groupes ont des propriétés relativement fortes) ou catégories (si la notion d'appartenance est plus faible, ou plus disparate).

La classification automatique apparaît soit comme un processus **supervisé**, c'est-à-dire avec un partitionnement préalable des documents en catégories ou classes, réalisé en général par un ou plusieurs experts humains servant de référence, soit comme un processus non supervisé, et là, l'apprentissage a toute latitude de faire émerger des regroupements à partir **de calcul de proximité**, ou d'algorithmes dits de clustering «Bien qu'il puisse apparaître dans la littérature une définition différenciée : le terme de classification est réservé au non supervisé et celui de catégorisation au processus supervisé »

De façon massive, la classification automatique de textes est un domaine où la notion de fréquence d'occurrence de mots, et celle d'apprentissage sont tellement prépondérantes que les chercheurs n'envisagent très souvent pas qu'il puisse en être autrement. Que les résultats produits soient directement liés à des calculs de fréquence d'occurrence de termes extraits [Salton et al 1975], ou à des justifications à fondement psycho-cognitif comme dans **LSA**. L'approche par mot est pratiquement la seule à tenir le haut du pavé.

Le problème de la classification est similaire à celui des autres domaines de la fouille de texte. Tout dépend de la tâche sous-jacente et de ses objectifs.

Ici plusieurs types de buts peuvent apparaître :

- ✓ optimiser la classification d'un corpus donné : ici les méthodes les plus dépendantes des spécificités du corpus sont les plus adaptées à ce genre de tâche. Le calibrage de la méthode par les données s'impose. Qu'il soit supervisé ou non, le processus de classification est forcément lié à un apprentissage, si ce n'est de toutes les nuances du corpus du moins de ses paramètres principaux.
- ✓ Catégoriser de grosses masses de documents en cherchant à s'adapter au mieux à des catégories existantes : les classes dominent ici les données et les méthodes devront plutôt chercher à reconnaître au mieux les caractéristiques de ces catégories dans l'ensemble des données fournies. Dans ce cas les méthodes supervisées sont plus adaptées et on cherchera surtout à optimiser la fonction de représentation des catégories.
- ✓ Découvrir de nouvelles catégories à partir d'un ensemble de documents : cet objectif rejoint d'une certaine façon les prémisses de l'extraction de connaissances. Ainsi, une connaissance régulièrement découverte dans une masse de données peut servir de catégorie pour classer, selon un point de vue, des textes. Dans ce cadre, les méthodes à apprentissage peuvent jouer un rôle de "découvreur" de catégories, et une situation semi-supervisée s'impose, car il faut en effet connaître les catégories existantes et leurs propriétés pour décider du caractère innovant de tel regroupement possible des données.

Dans de nombreux cas, les travaux de la littérature ont du mal à faire la différence entre la tâche de **catégorisation de documents** et **la tâche de segmentation**. En effet, les méthodes de clustering en particulier (recherche d'agrégation) traitent les ensembles de textes comme des "méta-textes" (un grand texte général issu de la concaténation) [Zhao et

Karypis, 2005]. Classer des textes revient alors à segmenter ce méta-texte en zones thématiques indexables par des catégories existantes ou émergentes.

Les défauts que l'on peut trouver à l'ensemble de ces approches relèvent essentiellement de l'ignorance, délibérée ou non, des qualités langagières que possède un texte. Ce dernier, comme nous l'avons dit, ne se comporte pas uniquement comme un ensemble de données. Un texte sur impose données et connaissances, signal et raisonnement. C'est ce qui fait qu'un texte relève d'un discours, et non pas de la concaténation de chaînes lexicales. Un texte est animé par une intention particulière. Dans ces conditions réduire le texte à ses termes constituants, et plus particulièrement à ses constituants dits significatifs (noms, verbes, adjectifs, adverbes dans les meilleurs des cas), c'est perdre une grande partie de l'intention communicative du discours. Les méthodes à forte dominance lexicale pourront effectivement classer un texte dans une catégorie, à condition dans ce cas que les catégories soient suffisamment disjointes lexicalement pour que le classement se fasse. Sans supervision, il sera difficile de mesurer les véritables performances de ce classement.

3.1 Les méthodes utilisées :

A. Classificateur bayésien naïf :

Comme son nom l'indique, ce classificateur se base sur le théorème de Bayes permettant de calculer les probabilités conditionnelles. Dans un contexte général, ce théorème fournit une façon de calculer la probabilité conditionnelle d'une cause sachant la présence d'un effet, à partir de la probabilité conditionnelle de l'effet sachant la présence de la cause ainsi que des probabilités a priori de la cause et de l'effet.

On peut résumer son utilisation lorsqu'il est appliqué à la classification de textes ainsi :

- ✓ On cherche la classification qui maximise la probabilité d'observer les mots du document.
- ✓ Lors de la phase d'entraînement, le classificateur calcule les probabilités qu'un nouveau document appartienne à telle catégorie à partir de la proportion des documents d'entraînement appartenant à cette catégorie. Il calcule aussi la probabilité qu'un mot donné soit présent dans un texte, sachant que ce texte appartient à telle catégorie.

- ✓ Par la suite, quand un nouveau document doit être classé, on calcule les probabilités qu'il appartienne à chacune des catégories à l'aide de la règle de Bayes et des chiffres calculés à l'étape précédente [Harry Zhang, 2004].

B. ***Algorithme des k-voisins les plus proches :***

L'algorithme des k-voisins les plus proches («*k-nearest neighbors*» ou kNN) est une méthode d'apprentissage à base d'instances.

La méthode ne nécessite pas de phase d'apprentissage, c'est l'échantillon d'apprentissage, associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

Lorsqu'un nouveau document à classer arrive, il est comparé aux documents d'entraînement à l'aide d'une mesure de similarité. Ses *k* plus proches voisins sont alors considérés : on observe leur catégorie et celle qui revient le plus parmi les voisins est assignée au document à classer. C'est là une version de base de l'algorithme que l'on peut raffiner. Souvent, on pondère les voisins par la distance qui les sépare du nouveau texte [Bentley, 1975].

Classification par SVM :

Les SVM, en anglais Support Vector Machine (Machines à Points de Support ou encore Séparateurs à Vaste Marge) sont reconnus pour leurs performances inégalées dans l'application à la classification de textes [Dumais et al. 1998]. De manière simplifiée, cette méthode consiste à rechercher un hyperplan séparateur pour deux classes données de manière à maximiser la marge entre les exemples de chacune des deux classes. Les SVM présentent de plus l'intérêt de formaliser le problème d'optimisation à partir seulement des produits scalaires entre objets et ainsi de se prêter à l'utilisation d'un noyau. Cette dernière technique permet de plonger les objets dans un espace de dimension éventuellement plus grande favorisant ainsi la possibilité de trouver un bon séparateur.

C. LSA :

L'analyse sémantique latente Le point de départ d'une analyse sémantique latente consiste en un tableau lexical qui contient le nombre d'occurrences de chaque mot dans chacun des documents, un document pouvant être un texte, un paragraphe ou même une phrase. Pour dériver les relations sémantiques entre les mots on remplaçant le tableau de fréquences original par une approximation qui produit une sorte de lissage des associations. Pour cela, le tableau de fréquences fait l'objet d'une décomposition en valeurs singulières avant d'être recomposé à partir d'une fraction seulement de l'information qu'il contient. Les milliers de mots caractérisant les documents sont ainsi remplacés par des combinaisons linéaires ou 'dimensions sémantiques' sur lesquelles peuvent être situés les mots originaux. Tant que les mots segments originaux sont positionnés dans cet espace sémantique, ce qui permet de mesurer leur proximité. Plus précisément, le sens de chaque mot y est représenté par un vecteur. Pour mesurer la similarité sémantique entre deux mots, on calcule le cosinus entre les vecteurs qui les représentent.

D. Probabilistic latent semantic analysis PLSA :

Le modèle probabilistic latent semantic analysis (PLSA) [Hofmann, 1999] est un modèle standard de la littérature pour la modélisation des collections de documents textuels. Ce modèle s'appuie sur la notion de données de co-occurrence. PLSA c'est une extension probabiliste du modèle LSI [Deerwester et al, 1990]. Avec ce modèle les documents est modélisée comme un ensemble de paires (d, w) où $d \in \{1,.., D\}$ est un indice de document et $w \in \{1,.., W\}$ est un indice de mot. Chaque document est représenté par une distribution de probabilité sur les K valeurs de la variable thématique latente α, et chaque valeur correspond à une distribution de probabilité sur l'ensemble des mots de la collection. Le processus génératif correspondant est le suivant :

- ✓ un document d est tiré suivant la probabilité P (α/d).
- ✓ une thématique est tirée suivant la probabilité P (w |d)

E. ***LDA :***

Latent Dirichlet Allocation (LDA), un modèle probabiliste génératif pour les collections de données discrètes comme corpus de textes. LDA est un modèle bayésien hiérarchique à trois niveaux, où chaque élément d'un ensemble est modélisé comme un mélange fini sur un ensemble sous-jacent de sujets. Chaque sujet est, en tourner, modélisée comme un mélange infini sur un ensemble sous-jacent de probabilités de sujet. Dans le contexte de modélisation de texte, les probabilités de rubrique fournissent une représentation explicite d'un document. Nous présentons techniques d'inférence approximatives efficaces basées sur des méthodes variationnelles et un algorithme EM pour l'estimation des paramètres bayésienne empirique. Nous présentons les résultats de la modélisation de documents, classification de texte, et filtrage collaboratif, en comparant à un mélange de modèle unigrammes et le LSI probabiliste modèle [David M. Blei, 2003].

4. La segmentation de textes :

La segmentation de texte est une tâche de reconnaissance thématique qui peut s'apparenter à une forme d'indexation. L'idée est de dégager des parties de textes offrant une certaine cohérence, et de les distinguer les unes des autres soit en les nommant (et du coup, cela indexe les parties en question) soit en délimitant les contours. Dans ce dernier cas, la segmentation de texte est assimilable à la détection des ruptures thématiques.

Certains travaux, comme ceux de Reynar [Reynar 1998], puis de Ji et Zha ont largement exploité l'analogie que l'on pouvait tirer de la métaphore de la reconnaissance d'image. Segmenter un texte ou reconnaître une image dans un ensemble complexe seraient des tâches proches. Deux attitudes sont alors possibles : soit on reconnaît ce qui est caractéristique de l'image (son centre, sa zone la plus typique, la zone la plus dense), soit on en détecte les contours. Ainsi, on peut segmenter un texte par extraction de parties typiques et, comme le font certains auteurs, mise en marge d'un index dans la marge. En revanche, cette structuration est floue sur les frontières. Les algorithmes de pavage de texte à la Hearst sont obligés de calculer des degrés de cohésion pour limiter l'extension des pavés ainsi reconnus.

On pourrait leur opposer des algorithmes de recherche des ruptures, qui tendraient à s'apparenter d'avantage à des algorithmes de contour.

Ce que pourraient avoir en commun toutes les recherches sur la segmentation de texte ce sont les caractéristiques suivantes :

1) La détection de la cohésion (thématique, lexicale) dans un texte
2) La définition de la limite de segment lorsqu'il y a rupture de cohésion : soit par changement lexical, soit par un éloignement sémantique autrement détecté
3) La capacité à présumer de l'unité du segment par rapport à une unité connexe et cohérente : ainsi, des segments seront constitués de plusieurs phrases adjacentes si toutefois ces dernières maintiennent la cohésion choisie. Des phrases séparées par plusieurs autres phrases ne pourront pas relever d'un même segment, sauf à considérer les phrases intermédiaires comme une forme remplissage (filler) qui ne rompt pas la chaîne (thématique, lexicale) ainsi créée.

Toujours est-il que les tâches de segmentation dépendent fortement de ce pourquoi elles sont réalisées. Elles peuvent être par exemple associées :

- à une tâche de recherche d'information, dans laquelle on cherchera à fournir en réponse non seulement un texte (issu d'une URL par exemple) mais plutôt, dans ce texte, le ou les fragments les plus véritablement compatibles avec la question posée.
- A une tâche d'indexation d'un texte pour des buts de création de métadonnées à usage pédagogique ou documentaire.
- A une tâche de résumé automatique ou semi-automatique, dirigé par le thème, où le résumé se fait par extraction des segments les plus appropriés à un thème donné, et création d'un nouveau document

A des travaux d'extraction du plan ou de la structure du document pour diverses fonctions ultérieures.

A. Les arbres de décision :

Les arbres de décision sont des méthodes de fouille de données qui relèvent de l'apprentissage supervisé et produisent des règles du type "si-alors". Le processus d'apprentissage consiste ensuite à déterminer la classe d'un objet quelconque d'après la valeur de ses variables. Les arbres de décision utilisent en entrée un ensemble d'objets (n-uplets) décrits par des variables (attributs). Chaque objet appartient à une classe, les classes étant mutuellement exclusives. Pour construire un arbre de décision, il est nécessaire de disposer d'une population d'apprentissage (table ou vue) constituée d'objets dont la classe est connue. Les méthodes de construction d'arbres de décision segmentent la population d'apprentissage afin d'obtenir des groupes au sein desquels la proportion d'une classe est maximisée.

Cette segmentation est ensuite réappliquée de façon récursive sur les partitions obtenues. La recherche de la meilleure partition lors de la segmentation d'un nœud revient à rechercher la variable la plus discriminante pour les classes. C'est ainsi que l'arbre (ou plus généralement le graphe) est constitué.

Finalement, les règles de décision sont obtenues en suivant les chemins partant de la racine de l'arbre (la population entière) jusqu'à ses feuilles [Breiman et all, 1984].

B. La méthode des k-means :

La méthode des k-means est un algorithme de classification automatique qui procède par réallocation dynamique [MacQueen 1967]. On l'appelle aussi la méthode des centres mobiles. En effet, il s'agit d'un algorithme itératif qui partitionne une population X en k classes les plus homogènes possibles où chaque classe est modélisée par son barycentre (c'est-à-dire la moyenne arithmétique de tous les individus affectés à la classe).

Il y a plusieurs versions qui sont améliorées par la suite par exemple *Fuzzy C-means, Sphérical k-means etc.*

- ***Critiques de la méthode :***

  - ✓ la méthode des k-means et ses variantes résolvent une tâche dite non supervisée, c'est-à-dire qu'elle ne nécessite aucune information sur les données. La segmentation peut être utile pour découvrir une structure cachée qui permettra d'améliorer les résultats de méthodes d'apprentissage supervisé (classification, estimation, prédiction).
  - ✓ Applicable à tous type de données : en choisissant une bonne notion de distance, la méthode peut s'appliquer à tout type de données (mêmes textuelles).
  - ✓ Facile à implanter : la méthode ne nécessite que peu de transformations sur les données (excepté les normalisations de valeurs numériques), il n'y a pas de champ particulier à identifier.

Les désavantages :

- Problème du choix de la distance : les performances de la méthode (la qualité des groupes constitués) sont dépendantes du choix d'une bonne mesure de similarité ce qui est une tâche délicate surtout lorsque les données sont de types différents.
- Le choix des bons paramètres : la méthode est sensible au choix des bons paramètres, en particulier, le choix du nombre k de groupes à constituer. Un mauvais choix de k produit de mauvais résultats. Ce choix peut être fait en combinant différentes méthodes, mais la complexité de l'algorithme augmente.
- L'interprétation des résultats : il est difficile d'interpréter les résultats produits, en d'autres termes, d'attribuer une signification aux groupes constitués. Ceci est général pour les méthodes de segmentation.

5. *Le profilage :*

Le profilage consiste à donner des contours lexicaux, sémantiques ou rhétoriques à un ensemble de textes ou de fragments de textes dans le but de :

- reconnaître un auteur particulier ou une période donnée dans une masse de documents non datés ou mélangés ces recherches d'identité ou de signature font du profilage une tâche particulièrement utile à des fins historiques ou culturelles
- à l'inverse, étant donné un profil fait de préférences fournies par des utilisateurs, rechercher l'ensemble des textes obéissant à ce profil : cela fait du profilage une forme intéressante de veille technologique, stratégique ou scientifique
- détecter des tendances ou des opinions dans des discours : la notion de jugement (favorable, neutre, défavorable) peut largement servir aux tâches prédictives du marketing ou des instituts de sondage. Mais également, des tendances plus complexes, de type attitudes majoritaires ou minoritaires, peuvent être l'objet d'une recherche de profil.

Dans beaucoup de cas, les méthodes ou les processus du profilage peuvent fortement ressembler à ceux de la catégorisation supervisée (si le profil est fourni) ou de la segmentation thématique. Le profil est récupéré par la détection de la rupture plus que par celle de l'émergence, cette dernière s'inscrivant en négatif par rapport aux critères de rupture. La différence est surtout dans l'intention : la catégorisation consiste souvent à se limiter à la relation d'appartenance. La segmentation thématique est dépendante de la notion de thème (ce dont on parle).

Le profilage est plus complexe car il introduit l'identification par la manière dont on parle. Cette manière définit la tendance parfois plus que l'objet même du discours.

IV. Conclusion :

Nous avons présentés dans ce chapitre les déférents taches de processus de fouille de texte avec les méthodes les plus connus et les plus utile pour l'extraction des connaissances.

Partie II

Conception Et Réalisation

Chapitre IV

Conception

I. Introduction :

L'agrégation de données textuelles permet de résumer le volume des données à visualiser lors d'une même analyse. En réduisant ainsi le volume des données par des méthodes de synthèse, l'utilisateur peut avoir une vision plus globale du domaine qu'il analyse.

Dans ce contexte nous proposons un processus d'intégration des données textuelles dans l'analyse OLAP. Il est basé principalement sur le clustring des documents.

II. Cas d'étude :

Notre cas d'étude est sur les blogs. Un blog est un type de site web, ou une partie d'un site web utilisé pour la publication périodique et régulière de nouveaux articles, généralement succincts, et rendant compte d'une actualité autour d'un sujet donné ou d'une profession. À la manière d'un journal de bord, ces articles ou « billets » sont typiquement datés, signés et se succèdent dans un ordre antéchronologique, c'est-à-dire du plus récent au plus ancien. Les appellations blogue ou cybercarnet sont également utilisées, notamment au Québec.

III. Solution proposée :

1. Cas d'utilisation du système :

Le **diagramme de cas d'utilisation** est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. La figure 8 présente les cas d'utilisations de notre système.

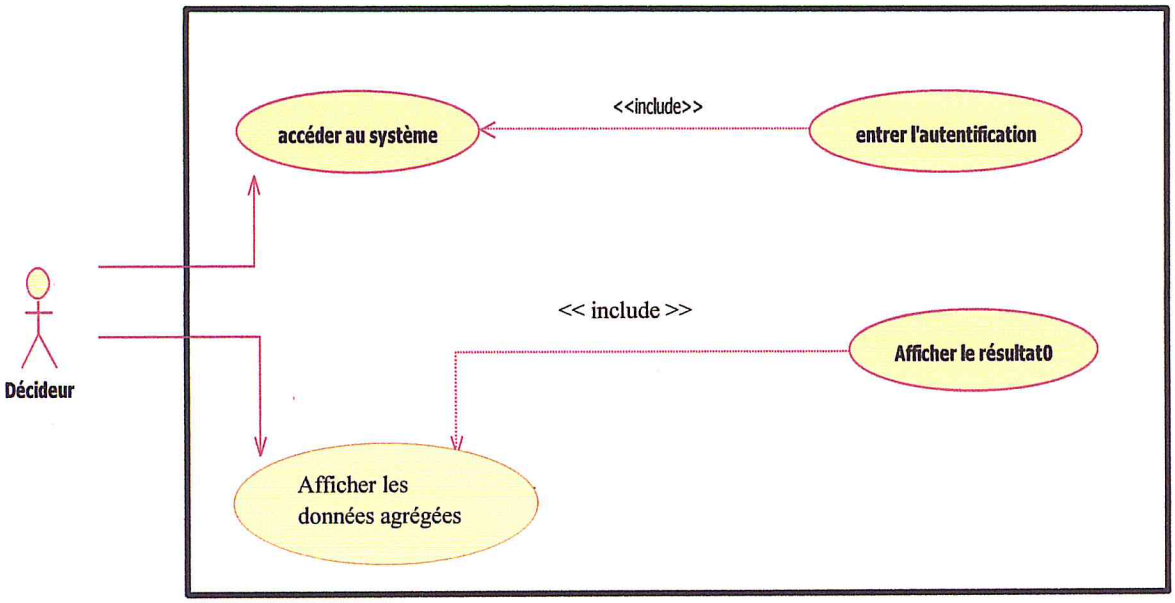


***Figure 8** : Diagramme de cas d'utilisation.*

2. La phase ETL :

2.1 Prétraitement :

Nous adoptons dans cette étape l'approche classique de fouille de texte qui combine les méthodes linguistique et statistique :

2.2 Suppression des mots outils :

Les mots qui composent un texte ne sont pas toujours importants. Il peut donc être intéressant de réduire le vocabulaire composant le texte, et pour cela il faut supprimer tous les mots vides comme les déterminants, prépositions, et les mots n'ayant pas de signification propre pour ne prendre en compte que les mots significatifs des textes, avec l'élimination des caractères spéciaux et des points de ponctuation.

2.3 Racinisation :

C'est la recherche de la racine des mots, aussi appelée racinisation, qui peut permettre un gain très important d'informations, pour les calculs effectués sur un corpus, et notamment dans les traitements appliqués aux mots directement. La lemmatisation permet de réunir des mots ayant la même racine, pour réaliser cette étape nous avons choisi d'appliquer l'algorithme de PORTER.

Algorithme de Porter :

L'algorithme développé par PORTER2 se compose de plusieurs règles de racinisation/désuffixation classées en sept phases successives (traitement des pluriels et verbes à la troisième personne du singulier, traitement du passé et du progressif,...). Les mots à analyser passent par tous les stades et dans ce cas plusieurs règles pourraient leur être appliquées, par exemple, "troubling" deviendra "troubl" par enlèvement du suffixe marqueur du progressif « **ing** » et sera ensuite transformé en "trouble" par application de la règle "bl" devient "ble" [3].

Algorithme :

R 1 est la région après le premier non-voyelle suivant une voyelle, ou à la fin du mot, si il n'y a pas une telle non-voyelle.

R 2 est la région après que le premier non-voyelle suite d'une voyelle dans *R* 1, ou à la fin du mot, si il n'y a pas une telle non-voyelle.

Un mot est appelé *court* si elle se termine par une syllabe courte, et si *R* 1 est nulle.

Si le mot a deux lettres ou moins, le laisser tel qu'il est.

Sinon, faire chacune des opérations suivantes :

Étape 0: Recherchez le plus long parmi les suffixes

'

's

's' éliminer si présents

Étape 1 ***a:*** Recherche pour la plus longue parmi les suffixes suivants, et effectuer l'action indiquée.

sses remplacé par **ss**

ied + ***ies*** replace par ***i*** si elle est précédée par plus d'une lettre,
sinon par ***ie*** (*ties* → *tie*, *cries* → *cri*)

s supprimer si la partie de mot précédent contient une voyelle pas immédiatement avant le ***s*** (*gas* et *this* conservent le ***s***, *gaps* et *kiwis* perdent)

us+ ***ss*** ne rien faire

Étape 1 ***b:*** Recherche pour la plus longue parmi les suffixes suivants, et effectuer l'action indiquée.

eed eedly remplacer par ***ee*** si en *R* 1

ed edly + ***ing ingly*** Supprimer si la partie de texte précédent contient une voyelle, et après l'effacement

Si le mot se termine ***at, bl*** ou ***iz*** ajouter ***e*** (*luxuriat* → *luxuriate),* ou

Si le mot se termine par un double supprimer la dernière lettre (*Hopp* →*hop),*

ou Si le mot est court, ajouter **e** (*hop* → *hope*)

Etape 1 ***c:***

Remplacer ***y*** ou ***Y*** par ***i*** si elle est précédée par une non-voyelle qui n'est pas la première lettre du mot (*cry* → *cri*, *by* → *by*, *say* → *say)*

Etape 2: Recherche pour la plus longue parmi les suffixes suivants, et s'il est reconnu et *R* 1, effectuer l'action indiquée.

tional: remplacer par ***tion***

enci : remplacer par ***rience***

anci : remplacer ***par*** l'ANCE

abli : remplacer par ***mesure***

entli : remplacer par ***ent***

izer , ization : remplacer par ***ize.***

ational , ation , ator: remplacer par ***ate***

alism aliti alli : remplacer par ***al***

fullness: remplacer par ***ful***

ousli, ousness : remplacer par ***ous***

iveness, iviti : remplacer par ***ive***

biliti, bli : remplacer par ***ble***

ogi : remplacer par ***og*** si elle est précédée par ***l***

fulli : remplacer par ***ful***

lessli : remplacer par ***less***

li : supprimer si elle est précédée par un valide ***li***

Etape 3:

Recherche pour la plus longue parmi les suffixes suivants, et s'il est reconnu et *R* 1, effectuer l'action indiquée.

tional : remplacer par ***tion***

ational : remplacer par ***ate***

alize : remplacer par ***al***

icate iciti ical : remplacer par ***ic***

ful ness : supprimer

ative : supprimer si en *R* 2

Etape 4:

al, ance ,ence , er, ic ,able, ible, ant, ement, ment ,ent ,ism ,ate ,iti ,ous ,ive ,ize supprimer

ion supprimer si elle est précédée par ***s*** ou ***t***

Etape 5:

Rechercher les suffixes suivants, et, s'il est détecté, effectuer l'action indiquée.

e supprimer si en *R* 2, ou *R* 1 et n'a pas été précédée par une courte syllabe
l supprimer si en *R* 2 et précédé par *l*
Enfin, tournez les ***lettres Y*** restants dans le mot nouveau dans minuscules.

3. ***Modèle de données :***

3.1 Schéma en étoile de notre cube de texte :

Le **modèle de données « en étoile »** est typique des structures multidimensionnelles stockant des données atomiques ou agrégées. Le modèle en étoile est souvent considéré (à tort) comme un modèle dénormalisé, ce qui permet une économie de jointures à l'interrogation. Il est ainsi optimisé pour les requêtes d'analyse. Il est implémenté sur un ***SGBD*** relationnel classique.

Le principe d'optimisation de ce modèle en étoile est le suivant : une clé calculée "technique" (clé générique) sert de jointure relationnelle entre les tables de dimensions et la table des faits. La requête ***SQL*** réalise d'abord sa sélection sur les tables de dimensions (peu volumineuses) et ensuite seulement, à partir des clés ainsi sélectionnées, la jointure avec la volumineuse table des faits

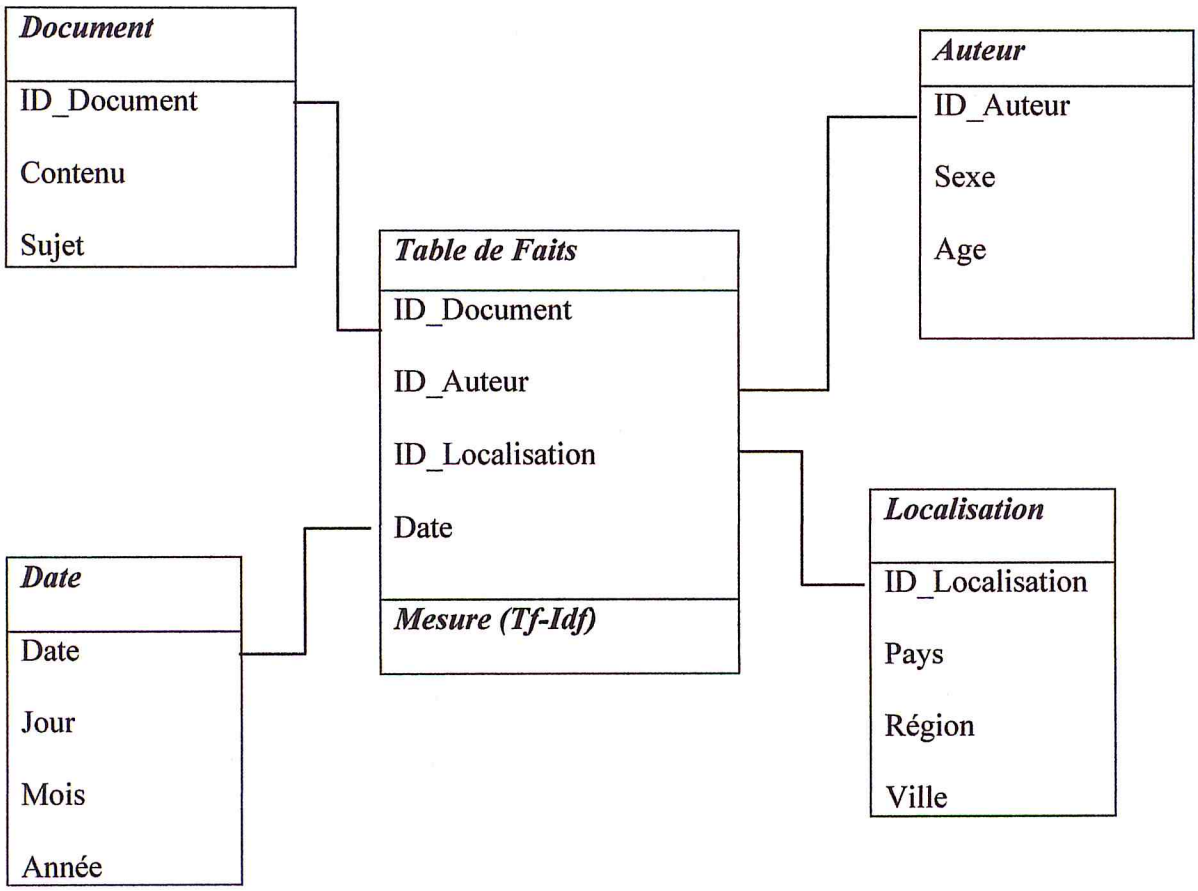


Figure 9 : *Schéma en étoile de notre cube de données.*

3.2 *Description des axes d'analyse:*

Les tables situées aux extrémités de l'étoile sont les axes d'analyse. Ce sont les **dimensions** explorées dans l'analyse.

- ***Document*** *(ID_Document, Contenu, Sujet)*

  ID_Document : Numéro séquentiel de document.

  Contenu *:* texte nettoyé

  Sujet : sujet de blog (Technology, Internet, Arts, …..)

- ***Auteur*** *(ID Auteur, Sexe, Age)*

  ID_Auteur : Numéro séquentiel de l'auteur.

  Sexe : male ou femelle.

  Age : c'est l'Age du blogueur.

- ***Localisation*** *(ID localisation, Pays, Région, Ville)*

  ID_localisation *:* Numéro séquentiel du pays.

  Pays *:* Pays de blogueur (USA, Angleterre, …..)

  Région : Cambridgeshire, Atlanta ….

  Ville *: Cambridge, Géorgie, ……*

  Un Auteur peut avoir plusieurs Localisations.

- ***Date*** *(Date, Jour, Mois, Année)*

  Date : dd/mm/yyyy

3.3 *Description de la mesure* d'analyse:

La table située au centre de l'étoile est la table de **fait.** Celle ci comprend des mesures : ce sont les éléments mesurés dans l'analyse comme les montants, les quantités, les taux, etc. et Dans notre cas (analyse des blogs), nous proposons d'utiliser la mesure Tf-Idf comme mesure d'analyse.

Table de faits *(ID_Document, ID Auteur, ID localisation, Date, TF_IDF)*

TF_IDF est la mesure de notre table de faits.

➢ *Représentation des documents par des vecteurs :*

Le problème de la représentation des documents consiste à trouver la meilleure représentation possible pour des textes faisant partie d'un corpus. Pour représenter un texte, nous pouvons opter pour une représentation simple et efficace. Dans le traitement de textes, la plupart des représentations se font à l'aide de vecteurs. Cette approche, appelée ***bag-of-words***, consiste à représenter un document par un vecteur de ***n*** termes pondérés pour chaque document. Le poids associé à chaque terme peut différer suivant les méthodes.

TF.IDF :

La méthode ***tf.idf*** (***term frequency inverse document frequency***) est très souvent utilisée pour attribuer des poids aux mots d'un corpus. L'idée est simplement de multiplier la *n* ième composante de chaque vecteur (qui est sa valeur "TF") par un coefficient qui dépend de la fréquence de l'unité correspondante dans l'ensemble des documents de ce corpus.

- La fréquence d'un terme (*term frequency*) est simplement le nombre d'occurrences de ce terme dans le document considéré.
- La fréquence inverse de document (*inverse document frequency*) est une mesure de l'importance du terme dans l'ensemble du corpus, calculée avec la formule suivante :

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Où

$|D|$: Nombre total de documents dans le corpus

$|D\|\{d_j : t_i \in d_j\}|$: Nombre de documents où le terme t_i apparaît (c'est-à-dire $n_{i,j} \neq 0$)

Finalement, le poids s'obtient en multipliant

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \cdot \mathrm{idf}_i$$

4. ***Agrégation de données textuelles :***

Pour l'analyse en ligne de données textuelles, nous proposons un couplage entre l'analyse en ligne OLAP et la fouille de données. Nous avons opté pour le clustring des documents la méthode spherical k_means.

4.1 Clustering (SPK_kmeans) :

Il s'agit de créer des groupes homogènes dans la population (l'ensemble des documents). Alors que notre choix de faire la segmentation des documents avec l'algorithme de spherical k_means avec lequel il est plus efficace d'exprimer la similarité entre les documents car il utilise la similarité cosinus au lieu de calculer la distance par les formules classique.

Algorithme :

Algorithm: spherical k-means (SPKM)
Input: A set of N *unit-length* data vectors $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ in $\mathbb{R}^d$ and the number of clusters K.
Output: A partition of the data vectors given by the cluster identity vector $\mathcal{Y} = \{y_1, ...y_N\}$, $y_n \in \{1, ..., K\}$.
Steps:

1. Initialization: initialize the *unit-length* cluster centroid vectors $\{\mu_1, ..., \mu_K\}$;
2. Data assignment: for each data vector $\mathbf{x}_n$, set $y_n = \arg\max_k \mathbf{x}_n^T \mu_k$;
3. Centroid estimation: for cluster k, let $\mathcal{X}_k = \{\mathbf{x}_n | y_n = k\}$, the centroid is estimated as $\mu_k = \sum_{\mathbf{x} \in \mathcal{X}_k} \mathbf{x} / \| \sum_{\mathbf{x} \in \mathcal{X}_k} \mathbf{x} \|$;
4. Stop if $\mathcal{Y}$ does not change, otherwise go back to Step 2a.

4.2 Mesure de similarité :

La **similarité cosinus** (ou **mesure cosinus**) permet de calculer la similarité entre deux vecteurs à dimensions en déterminant l'angle entre eux. Cette métrique est fréquemment utilisée en fouille de textes_[2].

Soit deux vecteurs A et B l'angle θ s'obtient par le produit scalaire et la norme des vecteurs : $\theta = \arccos\frac{A.B}{\|A\|.\|B\|}$

5. Thème d'un cluster (Top_Kw_Cluster):

Après la construction des clusters nous proposons d'agréger les informations de chaque cluster, par un thème représentatif des données textuelles (documents) que comporte ce cluster.

Un thème est représenté par les K mots clef les plus significatifs d'un cluster (Top_Kw_Cluster). Les K mots clef sont extraits de la manière suivante :

Pour chaque cluster :

- Recalculer les poids des mots clef par rapport aux documents existant dans le cluster, en utilisant les méthodes tf.idf.
- Extraire les k mots clef ayant les plus grand poids.

IV. Les diagrammes de séquence :

- ***Diagramme de séquence du cas d'utilisation "Accéder au système":***

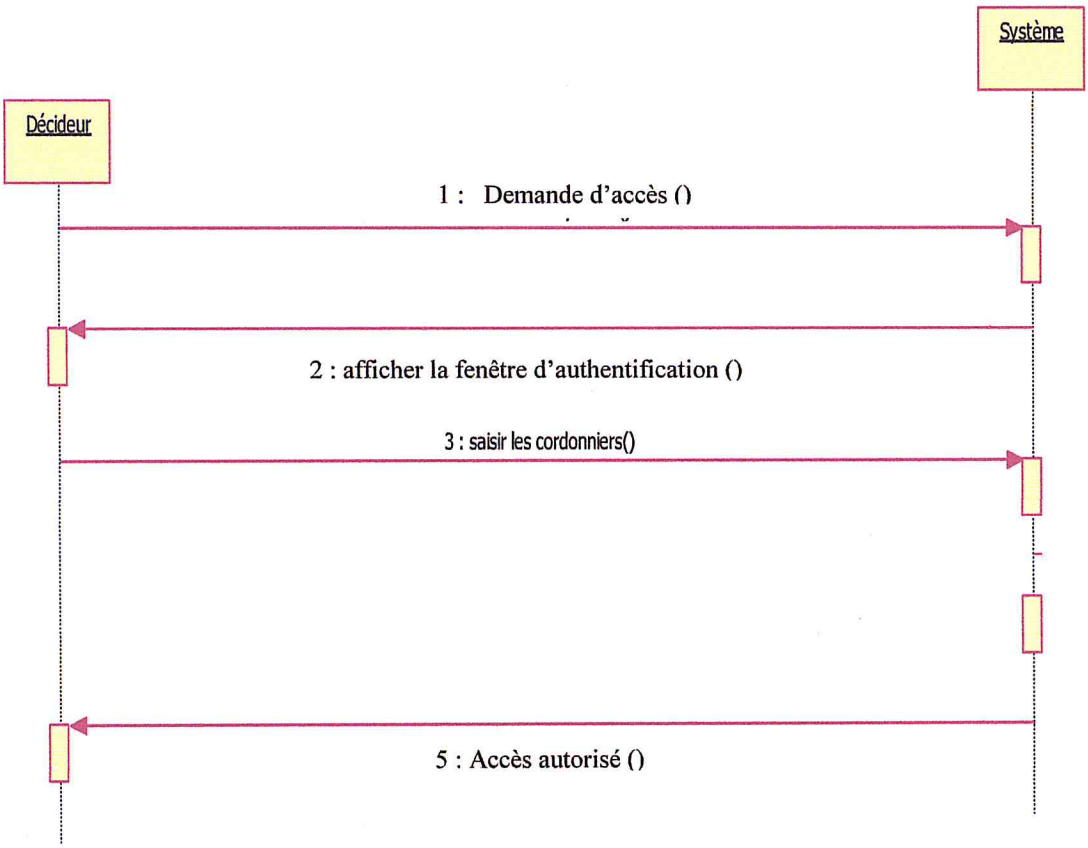


Figure 10 : *Diagramme de séquence du cas d'utilisation "Accéder au système"*

- ***Diagramme de séquence du cas d'utilisation "Afficher les données agrégées":***

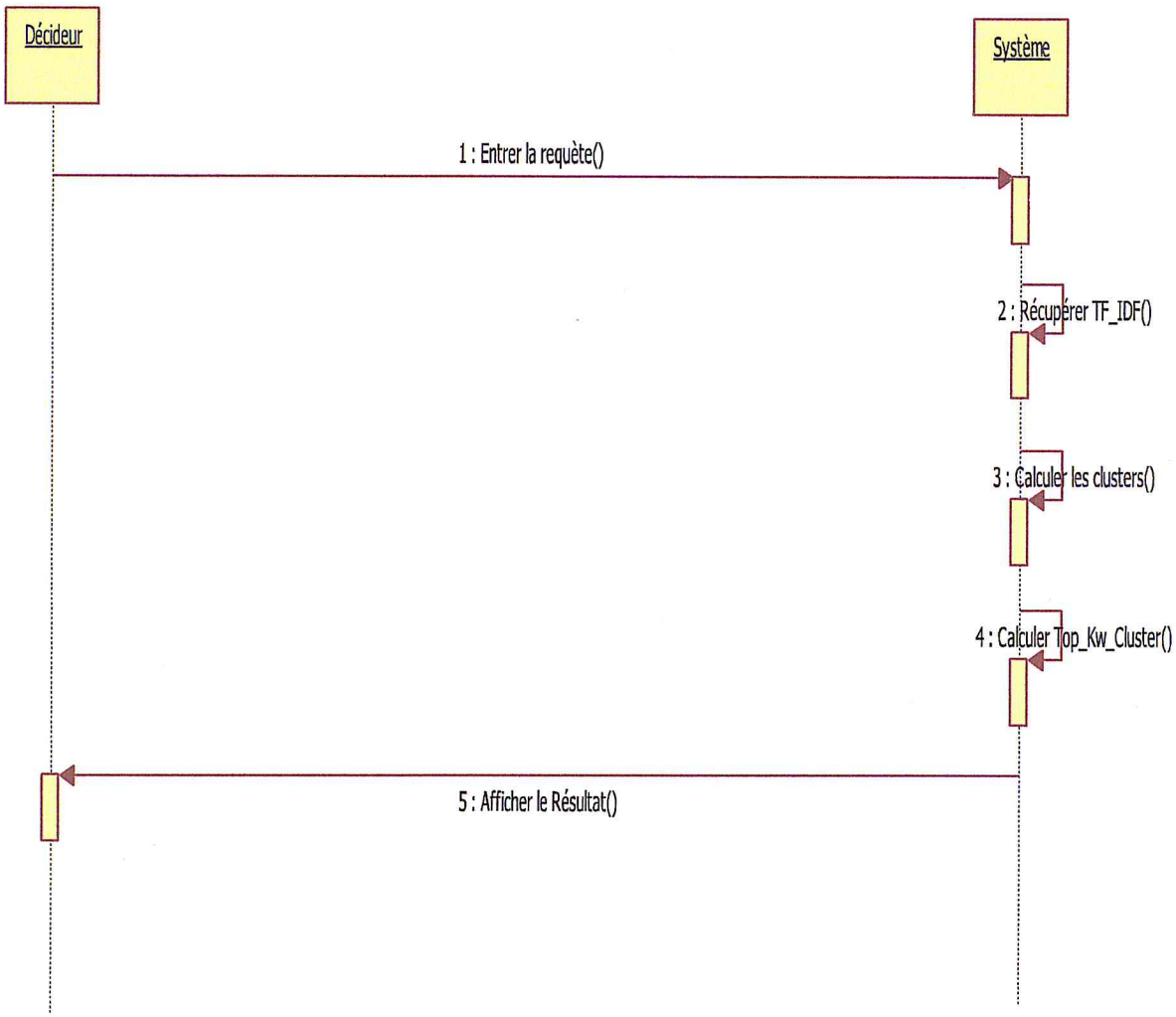


Figure 11: *Diagramme de séquence du cas d'utilisation "Afficher le données agrégées"*

Chapitre V

Implémentation de Système

I. Introduction :

Après l'expression des besoins et la modélisation du système, nous présentons dans ce chapitre les outils qui nous avons utilisés pour réaliser ce travail et sa mise en œuvre. Donc l'utilisation des outils de développement permettent de réduire considérablement la charge de travail, parmi ces outils on trouve les langages de programmation. Et pour les données l'utilisation des SGBD simples.

II. Jeu de données :

Nous avons utilisé un corpus de blog disponible en ligne [J.Schler et al, 2006]. Ce dernier est composé d'un grand ensemble de messages reçus de la part de 19 320 blogueurs qui sont réunis dans le site web [1] en Août 2004. Chaque blog est présenté comme un fichier séparé, et on trouve dans le nom de ce fichier le sujet de blog avec le sexe et l'âge du blogueur. Tous les blogueurs inclus dans le corpus sont dans l'un des trois groupes d'âge suivants : entre (13 ans, 17 ans), (23 ans, 27ans) ou entre (33 ans, 47 ans).Pour chaque groupe d'âge, il y a un nombre égal de blogueurs masculins et féminins Chaque blog dans le corpus comprend au moins 200 occurrences de mots anglais.

III. Présentation des outils :

1. Le langage de programmation (JAVA) :

Le langage utilisé pour l'implémentation du notre logiciel est le JAVA qui est un langage de programmation orienté objet, développer par SUN, le choix de ce langage est dû aux avantages suivants :

- Java est un langage simple à apprendre.
- Java est un langage orienté objet.
- Java est extensible à l'infini.
- Java est un langage à haut sécurité.

Mais pour avoir plus de confort il est préférable d'utiliser un environnement de développement éclipse.

De plus ce qui concerne notre projet, JAVA fournit tout ce qui est nécessaire à la manipulation des bases de données tels que :

- Etablir une connexion vers une base de données.
- Exécuter des requêtes à partir de code JAVA….etc.
-

2. *Système de Gestion de Base de Données(SGBD):*

Pour le chargement de notre entrepôt de données, nous avons choisi d'utiliser le SGBD relationnelle MySQL. Ce dernier fait partie fait partie des logiciels de gestion de base de données les plus utilisés au monde, c'est un logiciel libre qui développé sous double licence, produit libre, ou produit propriétaire.

Le serveur de gestion de base de données MySQL permet d'assurer :

- La définition et la manipulation des données.
- La cohérence des données.
- La sauvegarde et la restauration des données.
- La gestion des accès concurrents.

Pour l'interrogation de l'entrepôt de données, nous avons utilisées des requêtes SQL.

La figure 12 montre un exemple des données chargées dans notre entrepôt de données.

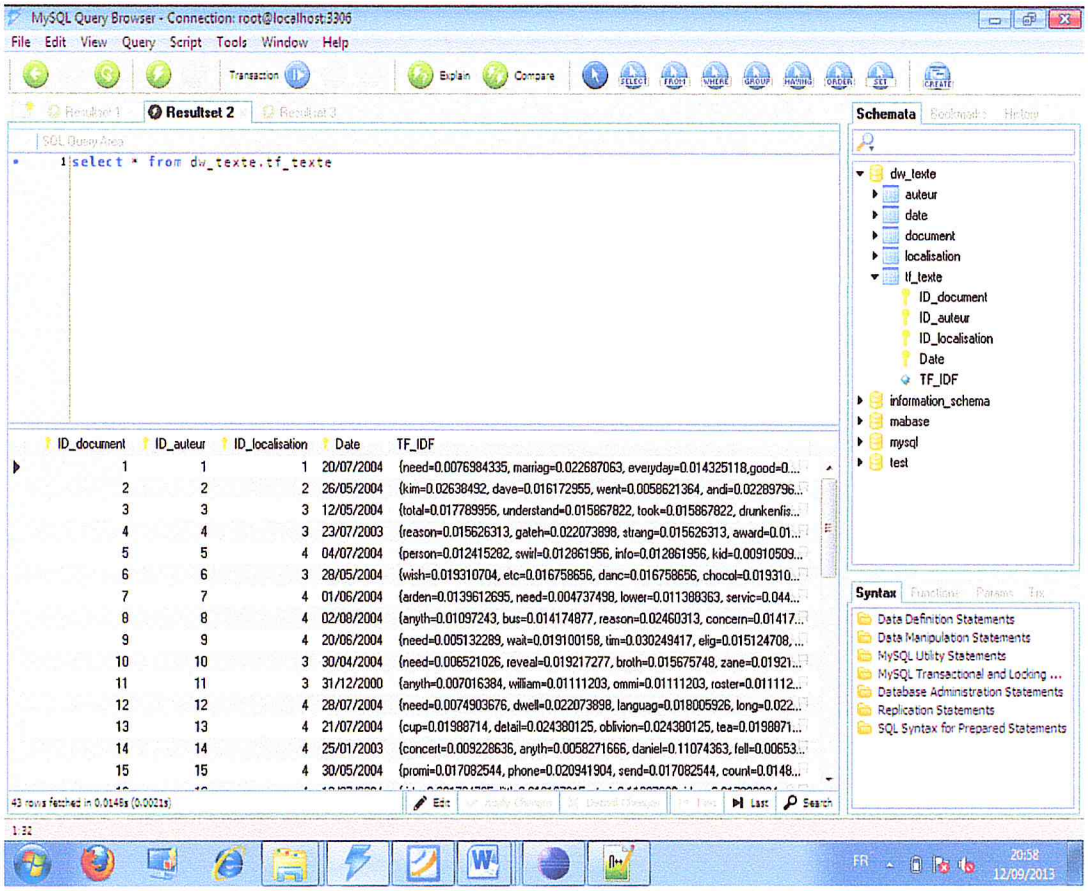


Figure 12 : *Exemple de Contenu de la Table de Faits*

IV. Les résultats :

- **Exemple de texte brut avant le prétraitement :**

« i've been trying to change the template for my blog for the past few days and i've been unsuccessful. it's very frustrating. i was able to do it once, but then when i wanted to change it again, it didn't work anymore. i did everything the same as that first time. it doesn't say anywhere that i can only change templates ONCE! does it? sigh. i better go check. »

- **le texte après le prétraitement :** (nettoyage des mots vides, points de ponctuation et la racinisation)

« tri chang templat blog past day unsuccess frustrat abl want chang work anymor everyth time anywher chang templat sigh better check »

- **Le Tf.Idf qui correspond ce texte est:**

{work=0.0328931, everyth=0.05730047, check=0.05730047, templat=0.1814965, unsuccess=0.07402436, abl=0.06424151, blog=0.047517624, past=0.047517624, tri=0.035192695, anymor=0.06424151, anywh=0.06424151, time=0.021010885, better=0.047517624, frustrat=0.06424151, sigh=0.07402436, day=0.027074467, chang=0.15574975, want=0.035192695}

- Accéder au système et enter l'authentification:

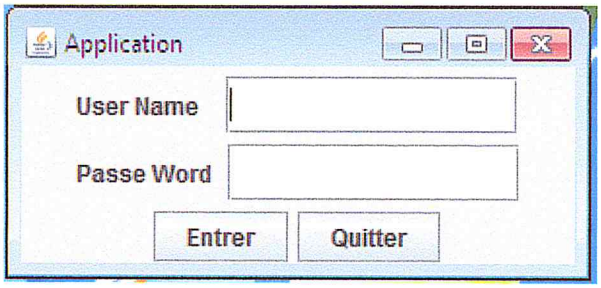


Figure 13 : *Fenêtre authentification*

- Enter la requête

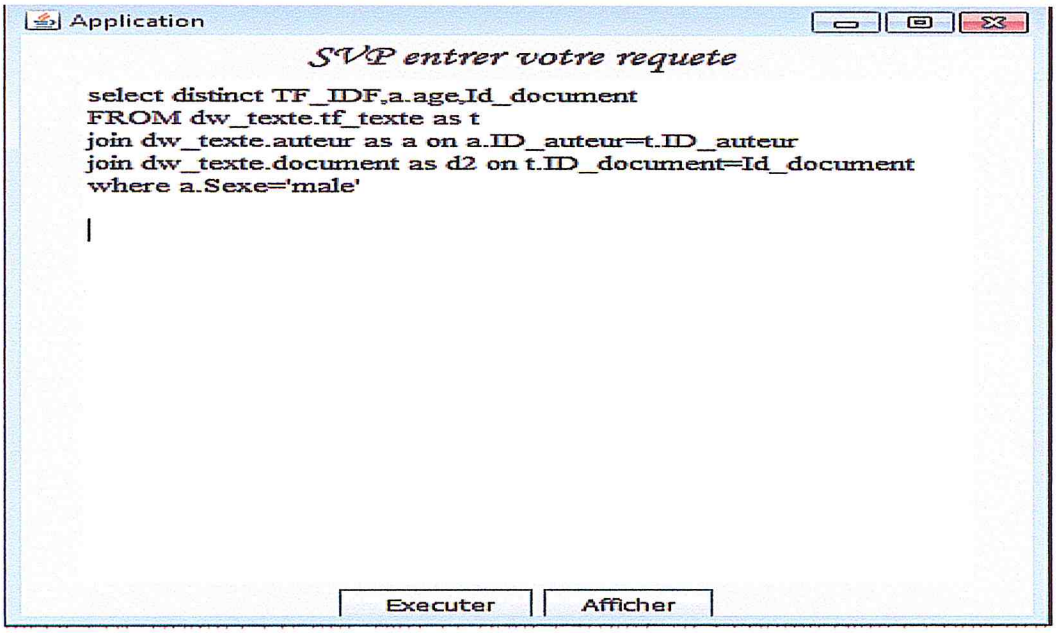


Figure 14: *Requêté d'analyse*

- Afficher les clusters :

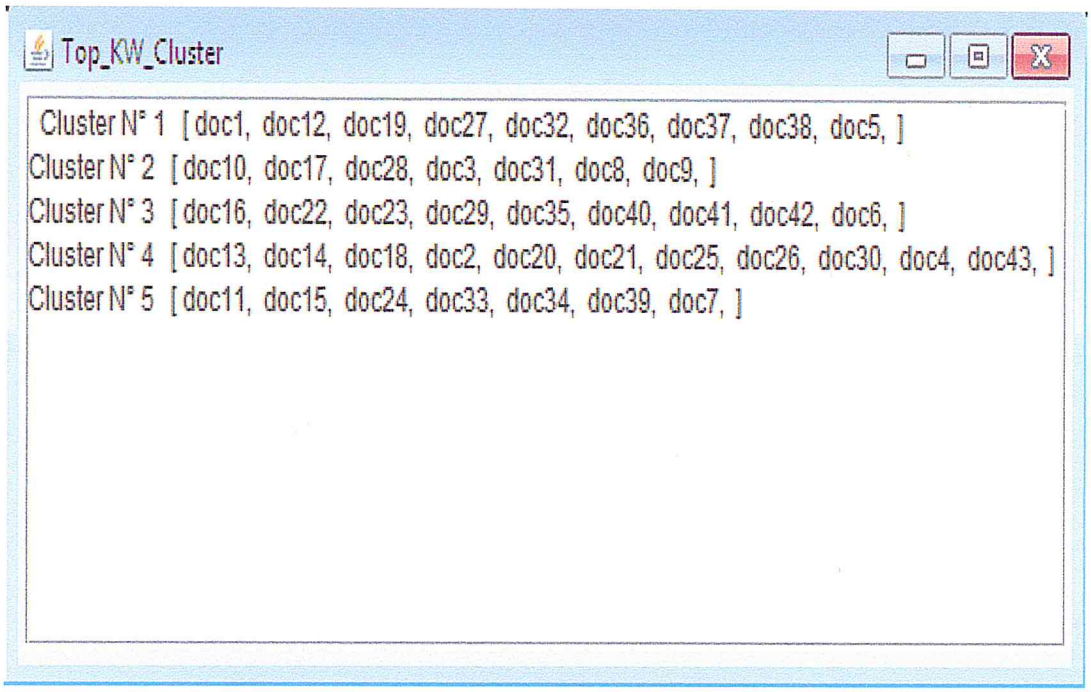


Figure 15 : *Les Cluster des documents*

Conclusion générale

Dans le cadre de ce mémoire, nous avons essayé d'apporter une solution au problème de l'analyse en ligne des données textuelles, à travers un couplage entre l'analyse en ligne et la fouille de texte.

L'opérateur d'agrégation « Tops_Kw_cluster » défini dans notre travail est basé sur une technique de segmentation de documents. Pour arriver à cette fin, nous avons effectué les tâches suivantes :

Premièrement, un prétraitement a été appliqué sur le corpus de texte (les blogs), avant d'être stocké dans l'entrepôt de données (la phase ETL). Dans cette phase, nous avons utilisé les méthodes classiques de prétraitement: nettoyage et racinisation…etc.

Ensuite, nous avons défini un modèle multidimensionnel adapté à l'analyse en ligne de données textuelles. Nous avons utilisé une représentation en schéma en étoile.

Enfin, après l'alimentation de l'entrepôt de données, nous avons utilisé la méthode de clustering « spherical k_means » pour agréger les données textuelles à partir de la requête d'analyse du décideur. Par la suite, nous extrayons les K mots clef les plus représentatifs de chaque cluster « Tops_Kw_cluster », ces mots clef représentent le sujet de cluster.

Ce projet était une expérience très intéressante dans le domaine décisionnel spécialement dans l'analyse en ligne des données textuelles, nous estimons que nous avons attend les objectives principaux défini auparavant.

Comme perspectives; une évaluation est requise pour mesurer les résultats d'analyse en ligne. Enfin, l'affichage des résultats dans notre application peut être amélioré par des techniques de visualisation plus élaboré.

BIBLIOGRAPHIE

[Ben-Messaoud et al, 2004] Messaoud R.B., Boussaid O., Rabas´ eda S., « A New OLAP Aggregation Based on the AHC Technique », in Proceedings of the 7th ACM International Workshop on Data Warehousing and OLAP (DOLAP'2004), pp. 65–72, Washington D.C., VA, USA : ACM Press. November 2004.

[Ben-Messaoud et al, 2006] Ben-Messaoud R., Boussaid O, Rabaséda S.L, « A Data Mining-Based OLAP Aggregation of Complex Data:Application on XML Documents », International Journal of Data Warehousing and Mining,2(4) :1–26. 2006.

[Ben-Messaoud, 2006] Riadh Ben Messaoud Couplage de l'analyse en ligne et de la fouille de données pour l'exploration, l'agrégation et l'explication des données complexes, Université Lumière Lyon 2, 2006

[Bentayeb et Favre, 2009] F. Bentayeb, C. Favre, « RoK: Roll-Up with the K-Means Clustering Method for Recommending OLAP Queries », Database and Expert Systems Applications (DEXA'09), pp.501-515, Linz, Austria, 2009.

[Bentley, 1975] Bentley, « *Multidimensional binary search trees used for associative searching* », *Comm. ACM 18, 509-517*, 1975.

[Benzécri, 1973] Benzécri J.P., L'analyse des correspondances, Paris : Dunold. 1973.

[Breiman et all, 1984] L. Breiman, J. Friedman, R. Olshen, C. Stone: *CART: Classification and Regression Trees*, Wadsworth International, 1984.

[Brill,1994] Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging.1994.

[David M. Blei, 2003] Journal de recherches sur l'apprentissage machine.David M. Blei, Michael JordanI. Andrew Y. Ng. *Université de Californie, Université de Stanford.2003*

[Dhillon, I. et al, 2001] Dhillon Concept decompositions for large sparse text data using clustering.Machine learning 42(1), (2001) ,143–175.

[Dumais et al. 1998] Dumais S, Platt J, Heckerman D, Sahami M. (1998).Inductive learning algorithms and representations for text categorization. Actes de CIKM '98, ACM Press, 148-155.

[Harry Zhang, 2004] Harry Zhang "The Optimality of Naive Bayes". Conférence FLAIRS2004.

[Hofmann, 1999] T.Hofmann, Probabilistic latent semantic analysis. In In Proc. of Uncertainty in Artificial Intelligence, UAI'99, pp. 289–296.

[Keith et al. 2005] Keith, S., O. Kaser, et D. Lemire Analyzing large collections of electronic text using olap. Technical Report TR-05-001, UNBSJ CSAS, 2005.

[Kimball R, 2000] Kimball R., Merz R. The data webhouse toolkit: building the Web-enabled data warehouse. 1re Èdition. John Wiley & Sons, 416 p. ISBN, 2000.

[MacQueen 1967] MacQueen, J. B. "Some Methods for classification and Analysis of Multivariate Observations". Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press. pp. 281297, 1967.

[Marlyse et Ghilani] Marlyse Dieungang – Khaoula Ghilani. *Datawarehouse: Cubes OLAP.*2005

[Mathieu Roche.2007] *Cours Fouille de Données. Mathieu Roche.2007.*

[Porter. 1980] M.F. Porter, (1980) "An algorithm for suffix stripping", Program: electronic library and information systems, Vol. 14 Iss: 3, pp.130 – 137.1980.

[Quinlan, 1986] R. Quinlan: *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., 1993.

[Ravat et al, 2007] Franck Ravat, Olivier Teste, Ronan Tournier, "OLAP Aggregation function for textual Data Warehouse", International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira - Portugal, 12-17 juin 2007, Vol. DISI, INSTICC Press, p. 151–156, juin 2007.

[REYNAR,1998] MEASUREMENTS OF THE PROTON AND DEUTERON SPIN STRUCTURE FUNCTIONS G_1 AND G_2, PHYS. REV. D 58, 112003 (1998) [54 PAGES].

[Salton et al 1975] Salton, G., A. Wong, et C. S. Yang A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 1975.

[Schler et al, 2006] J. Schler, M. Koppel, S. Argamon and J. Pennebaker (2006). Effects of [J. Age and Gender on Blogging in Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs

[Schmid, 1994] Schmid H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proceedings of the International Conference on New Methods in Language Processing, p. 44-49.

[Sebastiani, 2002] Sebastiani, F. Machine learning in automated text categorization. ACM Comput, 2002.

[Violaine et Yves, 2005] Le défi Fouille de Textes : Quels paradigmes pour la reconnaissance automatique d'auteurs ? Violaine Prince et Yves Kodratoff, LIRMM-CNRS et Université Montpellier.2005

[Yang et Chute 1992] Chute, C., Y. Yang "An overview of statistical methods for the classification and retrieval of patient events". Meth. Inform. Med., vol.34, pp.104-110, 1995.

[Zhang et al, 2009] Duo Zhang, Cheng xiang Zhai, Jiawei Han MITEXCUBE: Micro_Text Cluster Cube For On Line Analysis Of Text Cells.

❋ ***La liste des site Web :***

[1] http//www.crawdesign.com/business_intelligence/presentation.html.

[2] http://fr.wikipedia.org/wiki/Similarit_cosinus.

[3] http://snowball.tartarus.org/algorithms/english/stemmer.html.

[4] http//www.blogger.com.