Université de Montréal

BLED : Système d'aide à la recherche d'informations sur Internet

par

Kamal Bakour

Département d'informatique et
de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M. Sc.)
en informatique

Avril, 2005

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

BLED : Système d'aide à la recherche d'informations sur Internet

présenté par :

Kamal Bakour

a été évalué par un jury composé des personnes suivantes :

Gilles Brassard
Président-rapporteur

Esma Aimeur
Directeur de recherche

Philippe Langlais
Membre du jury

Mémoire accepté le

30 mai 2005

Sommaire

De nos jours, Internet connaît un développement phénoménal dans tous les domaines. Le moteur de recherche Google a recensé 4 285 199 774 documents en février 2004, et plus de 8 058 000 000 pages web de toutes catégories en novembre 2004 [34]. Ces chiffres illustrent bien l'ampleur de cette croissance. Néanmoins, l'inconvénient de cette révolution technologique réside dans le fait qu'il est difficile de délimiter l'information exacte que nous cherchons dans ces montagnes de données, qui contiennent naturellement des pages web redondantes, expirées ou parfois hors sujet.

Afin de pallier ces problèmes, il est absolument nécessaire de développer des outils permettant d'orienter et d'assister les internautes pendant leurs recherches sur la toile WWW [46].

Pour ce faire, nous proposons dans ce mémoire BLED (Best Links from Evaluated Data), un système facilitant la tâche de recherche aux internautes. L'approche prise par BLED est en fait une solution hybride ; utilisant d'une part une recommandation basée sur la similarité entre les mots clés, d'autre part une recommandation basée sur le regroupement par affinités. Celui-ci utilise la technique des règles d'association.

Mots clés : Forage de Données, Moteurs de Recherche, Recommandation, Classement, Règles d'Association, Forage du Web

Abstract

Nowadays, Internet experiences a phenomenal development in all fields like education, e-commerce, telecommunications, etc. For instance, Google search engine refers to 4 285 199 774 documents in February 2004, and more than 8 058 000 000 Web pages of all categories in November 2004 [34]. These values illustrate well the growth of Internet. Nevertheless the disadvantage of this technological revolution lies in the fact that it is difficult to situate exactly information that one is seeking on these mountains of data which naturally contain redundant, expired or sometimes out of topic web pages.

In order to overcome these problems, it is absolutely necessary to develop tools to direct and guide the users during their navigations in WWW (World Wide Web) [46]. In this thesis, we propose BLED (**B**est **L**inks from **E**valuated **D**ata), a new solution that makes easier the users's topic. The approach used in BLED is an hybrid solution, it uses at the same time two techniques; 1) recommendation technique based on the similarity between user's requests, and 2) the affinity grouping that uses Association Rules technique.

Keywords : Data mining, Search Engines, Recommendation, Page Ranking, Association rules, Web Mining .

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et des abréviations

Acronyme	Description	Première apparition
CART	Classification And Regression Trees	37
CBR	Case Base Reasoning	28
HTML	Hyper Text Meta Language	60
LDAP	Lightweight Directory Access Protocol	100
NCSA	National Center for Super computing Applications	56
OWL	Web Ontology Language	59
RDF	Resource Description Framework	59
TFIDF	Term Frequency Inverse Document Frequency	66
SQL	Structured Query Language	100
URL	Uniform Request Language	54
XML	Extend Market Language	59
HITS	Hyperlinked Induced Topic Search	3
SALSA	Stochastic Approach for Link-Structure	62
CLEVER	Client-Side EigenVector-Enhanced Retrieval	62
API	Application Program Interface	4
BLED	Best Links from Evaluated Data	4

Remerciements

Je voudrais en premier lieu adresser tout particulièrement mes vifs remerciements à ma directrice de recherche Madame Esma AIMEUR, qui m'a toujours orienté sur de précieuses pistes tout en me laissant totalement libre de choisir mes idées dans mes recherches.

Je tiens aussi à remercier tous les membres du jury, qui ont accepté d'être rapporteurs de mon mémoire.

Je remercie aussi tous mes collègues, que j'ai approchés dans le laboratoire HERON, qui m'ont aidé à réaliser ce modeste travail Nadjiba Djeddai, Arnoldo Rodriguez, Anita Salman et Sebastien Gambs
Je remercie également toute personne ayant contribué de près ou de loin à la réussite de ce travail, pour son soutien moral et technique.

Dédicaces

A mes parents, à mes frères et soeurs, et à toutes les personnes que j'estime.

Chapitre 1

Introduction

De nos jours, Internet est devenu l'un des moyens de communication les plus répandus dans tous les domaines, occasionnant une masse considérable de données et engendrant un volume important d'informations (archives, journaux électroniques, catalogues de bibliothèques, rapports techniques, articles, cours interactifs, films, images, sons, etc.), touchant différents domaines tels que l'éducation, la médecine, l'industrie, la sécurité, la défense, le commerce, etc.

Ces dernières années, de plus en plus d'internautes font confiance à Internet, surtout lorsqu'il s'agit de communiquer leurs informations personnelles à des sites web inconnus. Des systèmes de sécurité sont mis en place pour protéger ces internautes. De ce fait, il est possible d'acheter et de vendre des objets et d'utiliser les services offerts sur Internet qu'ils soient d'intérêt public, spécialisés ou commerciaux (universités, bibliothèques, banques, etc.). Ainsi, le nombre de sites web augmente sans cesse, chaque jour 60 Terabytes [10] de données sont ajoutées au WWW. Le moteur de recherche Google a annoncé récemment un chiffre effleurant les 8 milliards de pages web. Mais, une grande partie du web reste toujours

inaccessible, par exemple, des documents trop volumineux pour être entièrement indexés, des pages web protégées par l'auteur, des pages web dynamiques, etc. [34].

1.1 Problématique et motivation

Comme nous venons de le mentionner précédemment, Internet ouvre un grand portail pour accéder à diverses ressources telles que les documents scientifiques, les cours audiovisuels, les logiciels, etc. Cependant, jusqu'à présent, il n'existe aucun catalogue officiel, complet et mis à jour recensant toutes ces ressources mises à la disponibilité des internautes.

Toutefois, il existe des solutions multiples permettant d'accéder à une ressource sur Internet si nous ignorons l'adresse exacte du site web qui la détient. Par exemple, nous pouvons utiliser les moteurs de recherche comme Google[1], les méta-moteurs de recherche comme Eo[2], les annuaires de référence comme Yahoo![3], les portails spécialisés comme la toile du Québec[4] et les sites web commerciaux comme Hp[5], etc. Les adresses Internet de certaines ressources sont souvent connues en lisant un journal, un magazine, un article ou de bouche à oreille, ou encore grâce à un ami ou à une émission de radio.

D'une manière générale, si l'internaute ne connaît pas l'adresse exacte du site web possédant l'information qu'il recherche, il utilisera l'un des outils suivants :

---

[1] www.google.com

[2] www.eo.st

[3] www.yahoo.com

[4] www.toile.qc.ca

[5] www.hp.com

1. Les annuaires de référence

Dans les annuaires de référence (*Directory*), l'internaute doit d'abord classer sa recherche dans un groupe qui décrit au mieux sa recherche, par exemple : "Hockey" dans la catégorie "Sport", ce qui n'est pas toujours facile. Une fois le classement fait, il parcourt l'arborescence des sites de cette catégorie proposés par l'annuaire de référence (Yahoo! par exemple). L'avantage des répertoires réside dans leur facilité d'utilisation quand les sujets recherchés sont faciles à classer. En revanche, le problème se manifeste grandement lorsqu'il est difficile de déterminer à quelle catégorie appartient une recherche.

2. Les moteurs de recherche

La philosophie des moteurs de recherche est complètement différente de celle utilisée dans les annuaires de référence. Ici, l'internaute interroge un moteur de recherche (Google, Altavista, Lycos, Teoma, etc.) via son interface graphique, en utilisant un ensemble de mots clés décrivant sa recherche. Cependant, lorsque cet internaute envoie sa requête, il risque de se confronter à des milliers de pages web à explorer. Certaines sont parfois redondantes, expirées où ne répondant même pas aux critères de sa recherche. Dans la majorité de ces cas, nous pensons que l'utilisateur ne consulte que les 2 ou 3 premières pages web apparaissant à l'écran, ensuite il change les mots clés de sa requête. Ce problème demeure malgré les efforts d'amélioration des moteurs de recherche au niveau de la pertinence des résultats, Pourtant ils utilisent des algorithmes de classement extrêmement sophistiqués tels que : PageRank [58] utilisé par Google ou HITS [52] utilisé par CLEVER.

Cette faiblesse ressentie au niveau des moteurs de recherche nous a motivé à chercher des solutions ou des méthodes qui diminueraient l'intensité du problème.

1.2 Objectifs

Le sujet de recherche que nous proposons consiste en la conception et la réalisation d'un système d'aide à la recherche d'informations sur Internet. Ce système utilise la base de données du moteur de recherche Google comme source d'information. Ce choix réside dans le fait que Google est considéré comme étant le plus puissant moteur de recherche, de plus, sa base de données est très riche en termes de qualité et est accessible à travers ses API (Application Program Interface) .
La solution que nous suggérons est une alternative aux autres moteurs de recherche traditionnels, mais avec de nouvelles fonctionnalités. Son objectif est d'aider et d'assister l'internaute à tirer profit d'Internet en se basant sur des nouveaux aspects. En effet, avec notre système les utilisateurs pourraient économiser le temps consacré pour la recherche, en évitant le problème dû au parcours des pages web inutiles, et en profitant naturellement des recherches effectuées par d'autres utilisateurs, entre autres, lorsqu'ils n'ont pas suffisamment le temps nécessaire pour explorer un grand nombre de pages web.
Les utilisateurs de notre système peuvent également acquérir des connaissances sur l'élaboration de mots clés en visant précisément les informations qu'ils cherchent. La difficulté de l'utilisation des moteurs de recherche réside principalement dans la façon de choisir les mots clés qui correspondent aux informations recherchées. En outre, notre système est capable de recommander et de partager les meilleurs documents entre internautes en utilisant le forage de données.
Pour atteindre ces objectifs, notre système nommé BLED (**B**est **L**ink from **E**valuated **D**ata) utilise une solution mixte : 1) une recommandation basée sur la similarité entre les requêtes des utilisateurs ; 2) une technique de forage de données[6]

[6]Data Mining

appelée *règles d'association* permettant de fournir des résultats plus raffinés aux utilisateurs en se basant sur les similarités possibles dans leurs historiques.

1.3 Plan du mémoire

Après cette brève introduction de la problématique, de nos motivations et de nos contributions. Le reste de ce mémoire est divisé en six chapitres.

Au **chapitre 2**, nous abordons des généralités sur les outils de recherche traditionnels, puis nous évoquons le problème de classement des pages web[7], nous présentons ensuite l'algorithme de classement PageRank, un algorithme utilisé par Google pour le classement de ses pages web.

Dans le **chapitre 3**, nous décrivons le forage de données, ses tâches, ses objectifs, son processus de déroulement et quelques techniques, notamment la technique des règles d'association.

Au **chapitre 4**, nous terminons l'état de l'art, en exposant le forage du web[8], une branche particulière de forage de données spécifique aux données d'Internet.

Le **chapitre 5** décrit en détail l'architecture globale de système BLED, ses composants, ses techniques et ses algorithmes.

Par la suite, au **chapitre 6**, nous montrons l'implémentation et l'expérimentation du système BLED.

Enfin, le **chapitre 7** clôture ce mémoire par une conclusion et une discussion, en présentant les perspectives de recherche pour notre système BLED.

---

[7]Page Ranking

[8]Web Mining

Chapitre 2

Outils de recherche

Nous présentons dans ce chapitre l'architecture et le fonctionnement des outils de recherche, puis nous parlerons de l'algorithme de classement PageRank [25] utilisé par le moteur de recherche Google pour calculer la popularité des pages des sites web. Ensuite, nous présenterons deux outils de recherches avancés situés dans notre contexte.

2.1 Généralités

Les outils de recherche sur Internet sont des logiciels qui permettent d'assister l'internaute à trouver les informations désirables, en utilisant des requêtes simples ou composées, écrites en langage naturel et en langue multiple. Lorsqu'un internaute, utilisant un outil de recherche (Google, Yahoo!, etc.), soumet sa requête pour chercher une information concernant un sujet quelconque, cet outil lui proposera en interrogeant sa base de données locale une liste contenant des ré-

sultats (URLs) qui correspondent le mieux à sa requête. Ces résultats dépendent naturellement de certains éléments comme l'architecture et la technologie utilisées par l'outil (moteur de recherche, annuaire de référence, portail, méta-moteur de recherche). Pour une même recherche, il est évident que la liste des résultats retournés par le moteur de recherche Google n'est probablement pas identique à celle retournée par le répertoire Yahoo ! ou le moteur de recherche Altavista. En outre, la façon d'élaborer des requêtes est également un facteur important dans la recherche.

2.2 Catégories des outils de recherche

Les outils de recherche sur Internet se scindent en deux grandes catégories : les annuaires de référence[1] et les moteurs de recherche[2].
Les portails et les méta-moteurs de recherche sont issus de ces deux catégories ; ils ne seront pas décrits dans ce mémoire.

2.2.1 Les annuaires de référence

Ce sont des sites web comme Yahoo ! et Voila qui référencent des services ou pages web sur Internet accessibles au moyen de liens hypertextes, classés en plusieurs catégories comme : voyage, santé, informatique, divertissement, art, culture, etc. Ils sont alimentés manuellement par une demande de référencement auprès de propriétaires des sites web. Leurs avantages est qu'ils sont faciles à utiliser et profitables pour les utilisateurs inexpérimentés sur des sujets vagues,

[1] Directory
[2] Search Engine

car ils référencent les principaux sites répondant à ces sujets. Néanmoins, leur véritable inconvénient est que leur mise à jour se fait manuellement.

2.2.2 Les moteurs de recherche

Contrairement aux annuaires de référence, les moteurs de recherche quant à eux possèdent des robots[3] intelligents qui parcourent les pages des sites web afin de les enregistrer, les indexer et les mettre à jour de façon automatique. Grâce à ces robots, les moteurs de recherche peuvent indexer les sites web de la toile **(WWW)**, et ils deviennent très efficaces et pratiques pour des recherches précises. Par exemple, pour un mathématicien qui recherche l'expression "**équations linéaires**", il est préférable qu'il aille consulter des sites web réservés pour les Maths s'il en connaît, ou un moteur de recherche qui peut lui recenser les pages web comprenant les mots clés de sa requête, que d'aller explorer sur un annuaire une grande arborescence comme : "**Sciences/Maths/Algèbre/.../Equations linéaires**".

Toutefois, l'inconvénient majeur de ces engins de recherche est qu'ils donnent trop de réponses et parfois ces réponses sont inutiles.

Parmi les moteurs de recherches les plus populaires, nous citons : Google, Hotbot, Lycos, Msn search, Excite, Fast, Iwon, Teoma, Yahoo, Altavista, Askjeevs, Wisenut [68, 69]. Pour plus de détails, voir l'annexe B.

Le tableau 2.1 dévoile la corrélation qui se manifeste entre les moteurs de recherche les plus utilisés sur Internet. Chaque ligne de ce tableau indique le nom d'un moteur de recherche, et chaque colonne détaille la relation qui existe entre celui-ci et d'autres moteurs de recherche. Par exemple, le moteur de recherche

[3]Crawlers, Spiders : Araignées de balayage.

AOL Search utilise Google pour ses "principaux résultats", AdWards de Google pour ses "liens promotionnels" et Open Directory (DMoz) pour ses "liens répertoriés". C'est-à-dire, AOL Search peut recenser tout site web soumis sur Google ou Open Directory (DMoz).

TAB. 2.1 – Relation entre les moteurs de recherche [69]

Moteur de Recherche	Type	Principaux résultats	Liens promotionnels	Liens répertoriés
AllTheWeb	Crawler	Yahoo	Overture	non
AltaVista	Crawler	Yahoo	Overture	Open Directory
AOL Search	Crawler	Google	Google	Open Directory
Ask Jeeves	Crawler	Teoma	Google	Non
Gigablast	Crawler	Gigablast	non	non
Google	Crawler	Google	Google	Open Directory
Msn Search	Crawler	Yahoo	Overture	non
Netscape	Crawler	Google	Google	Open Directory
Teoma	Crawler	Teoma	Google	non
Yahoo	Crawler	Yahoo	Overture	Yahoo

2.3 Architecture et fonctionnement des moteurs de recherche

D'une façon générale, un moteur de recherche fonctionne en trois étapes parfaitement structurées [6]. Tout d'abord, il amorce des robots [19] qui parcourent le réseau Internet dans le but d'acquérir le plus grand nombre possible de pages web. Il entreprend ensuite le processus d'indexation pour en construire la base de données d'index. Nous trouvons dans cette étape, par exemple, le découpage des documents en structures, en mots ou en thèmes. Et enfin, un moteur de recherche possède des algorithmes de tri et de comparaison lui permettant d'interroger sa base de données d'index pour constituer les résultats correspondant aux mots clés recherchés par les internautes pendant leurs recherches. Soulignons que la

pertinence des résultats retournés par les moteurs de recherche dépend fortement de ces trois composants ("Araignées de balayage", "processus d'indexation" et "programmes de recherche") [21]. La figure 2.1 illustre l'architecture générale des moteurs de recherche ainsi que la communication entre ces trois principales étapes.

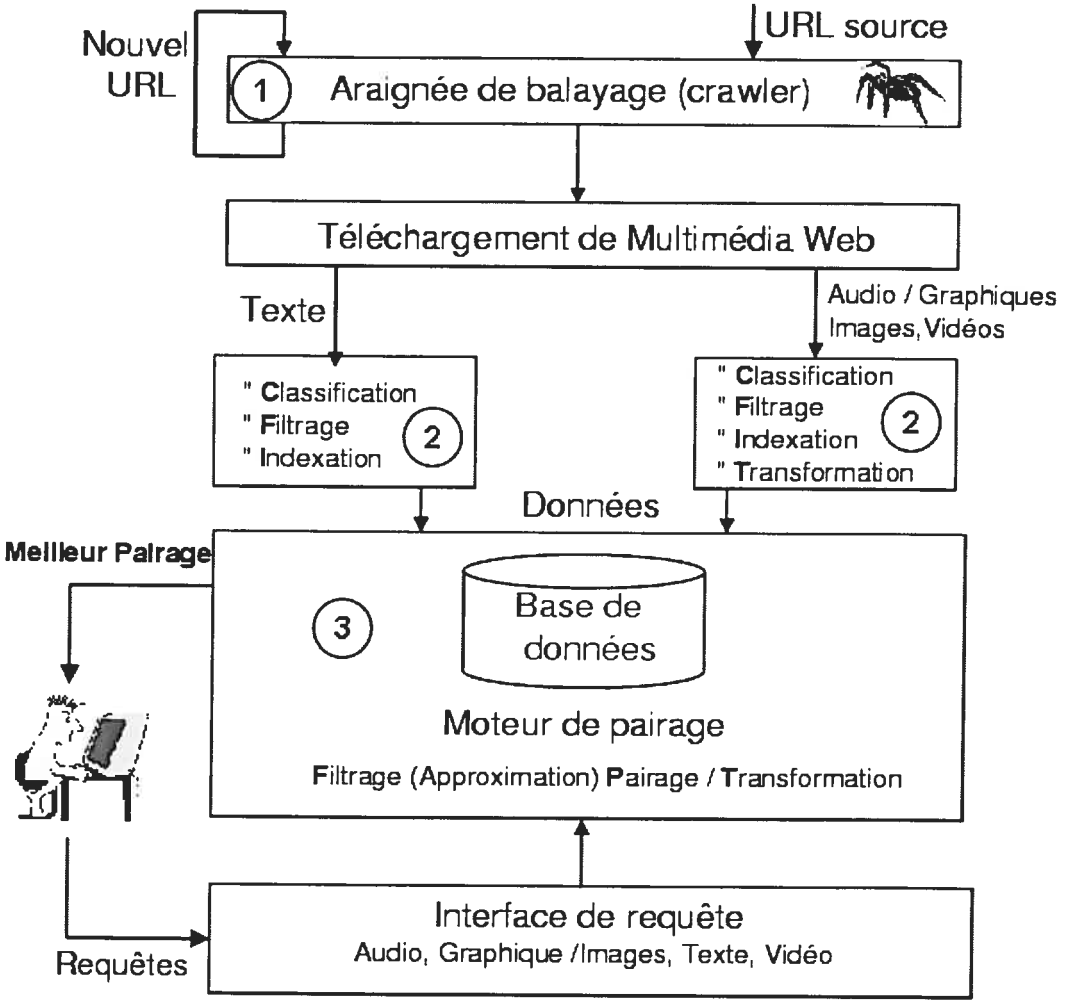

FIG. 2.1 – Architecture des moteurs de recherche [41, 40]

2.3.1 Les robots

Un robot [41, 40, 60, 66] est un programme informatique qui examine automatiquement l'ensemble des pages d'un site web afin de collecter ses ressources (documents textes, images, audio, vidéos, etc.) en commençant par sa racine lien par lien jusqu'à la fin de façon récursive. Ces ressources seront transmises ensuite au processus d'indexation afin de les filtrer, les transformer, les indexer et les classifier (voir figure 2.1 (1)).

2.3.2 Le processus d'indexation

Une fois les ressources bâties par les araignées de balayage, le processus d'indexation [41, 40, 66] les analysera pour en extraire certaines informations importantes. À titre d'exemple, le titre de la page web téléchargée, les textes délimités par les balises <BODY>, les informations contenues dans les balises <HEAD>, les images et les vidéos. Ces informations participeront à la construction de la base de données d'index, qui sera utilisée par le programme de recherche (voir figure 2.1 (2)) à l'étape suivante.

2.3.3 Le programme de recherche

Ce programme [41, 40, 66] sert à interroger la base de données d'index afin de constituer des listes de réponses, triées (classées) convenablement selon les mots clés introduits par les utilisateurs (voir figure 2.1 (3)).

2.3.4 Le classement et l'algorithme PageRank

Depuis longtemps, les créateurs des moteurs de recherche travaillent sans cesse pour améliorer la pertinence des résultats proposés aux clics des utilisateurs [22]. Algorithmiquement, la pertinence des résultats demeure difficile à résoudre, car il ne s'agit pas de calculs numériques précis, mais d'une tâche subjective et relative aux personnes. Citons l'exemple d'une personne qui recherche le mot **"jaguar"**, le moteur de recherche ne comprend pas exactement ce que cette personne attend comme réponse, c'est-à-dire, s'agit-t-il **de voitures jaguar** ou **de l'animal jaguar** ? Dans ces conditions, le moteur de recherche lui suggère tous les liens qui ont un rapport avec le mot **"jaguar".** Une solution à ce problème consiste à consolider cette recherche avec d'autres termes (mots clés) qui donnent plus de précision à cette requête. Google a trouvé un remède au problème de Ranking[4] grâce à ses fondateurs, Sergey Brin et Larry Page, étudiants à l'Université de Standford aux Etats Unis. Ces étudiants ont inventé un algorithme de classement très puissant appelé**"PageRank"**[16, 58], fondé sur les chaînes de Markov. Cet algorithme a positionné Google en tête des moteurs de recherche actuels.
PageRank est un algorithme **itératif** convergeant vers une valeur fixe après un **certain nombre d'itérations**. Il sert à mesurer l'importance ou la popularité d'une page web. Cette notion va permettre de determiner l'ordre de tri des réponses apparaissant à l'écran de l'internaute pendant qu'il fait des recherches.
Avant de présenter l'algorithme PageRank, il est préférable de définir les différents types de liens que nous pouvons trouver dans la toile (WWW). D'une manière générale, nous distinguons trois types de liens : internes, entrants et sortants.

[4] Ranking : classement

a) Liens internes

Ces liens forment la topologie d'un site web.

b) Liens entrants

Ces liens proviennent de sites extérieurs vers un site web.

c) Liens sortants

Ces liens indiquent les pages d'un site web qui pointent vers d'autres sites web.

La figure 2.2 montre un site web composé respectivement de 6 pages web internes : A, B, C, D, E et F. De plus, il y a deux pages web provenant de l'extérieur qui pointent sur ce site, lesquelles sont dénotées par E1 et E2 (liens entrants). Et enfin, ce site pointe sur des pages web externes, lesquelles sont désignées par S1, S2, S3 et S4 (liens sortants).

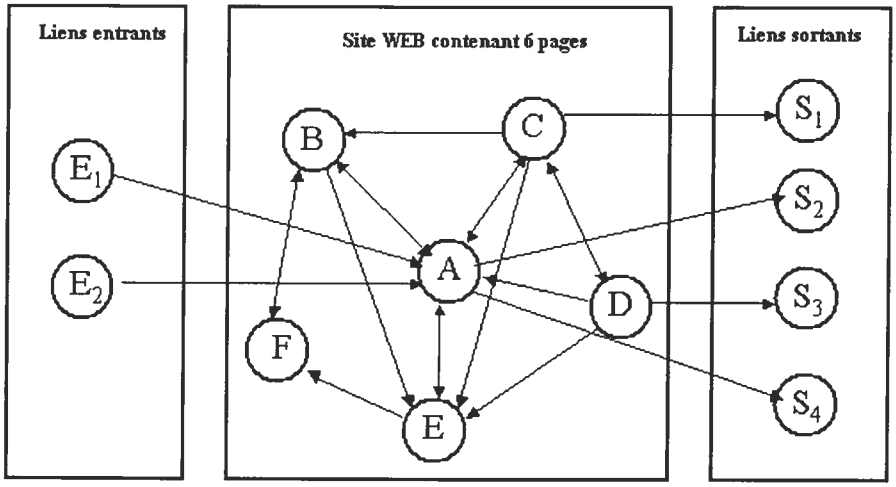

FIG. 2.2 – Type de liens

Maintenant, voici comment se présente la formule de l'algorithme PageRank [58]

$$PR(u) = (1 - r) + r * (\sum_{i=1}^{i=n} \frac{PR(v_i)}{|v_i|}) \tag{2.1}$$

Où

$PR(u)$: le PageRank de la page web u.

$|v_i|$: le nombre de liens qui pointent vers la page web v_i.

n : le nombre de liens qui pointent vers la page web u.

r : le coefficient d'amortissement qui est fixé à 0.85 par défaut.

Exemple

Reprenons le site web de l'exemple précédent pour mieux comprendre le fonctionnement de l'algorithme de PageRank. Rappelons que l'objectif de cet algorithme est de déterminer le **PR** d'une page web, qui permet de déterminer son importance ou sa popularité, nous signalons que les **PR** des liens entrants $E1$ et $E2$ sont initialisés par défaut à 0.85, car nous ne connaissons pas leurs valeurs exactes. Après avoir réitéré l'algorithme PageRank 25 **itérations**, nous avons pu constater que le PR des pages A, B, C, D, E et F est constant (voir tableau 2.2). L'ordre de l'importance de ces pages web se fait en effectuant le classement de ces pages web selon l'ordre décroissant de leur **PR**. Dans cet exemple, l'ordre d'importance est alors : A, B, E, F, C et D. Cette constatation semble très juste et raisonnable en même temps, de fait que la page web A reçoit plus de liens provenant de l'extérieur par rapport aux autres pages web B, C, D, E et F. De plus, la popularité d'une page web dépend du nombre de liens émis sur elle. Il est clair que la page web principale de Sun[5] est très populaire, car elle est référencée partout à travers les annuaires et les moteurs de recherche et sur des sites commerciaux et personnels.

[5] www.sun.com

TAB. 2.2 – Exemple de calcul de PageRank

Itération/PageRank	1	2	3	4	5	...	25
PR(A)	2.940	3.099	3.141	3.162	3.173	...	3.187
PR(B)	1.669	1.834	1.878	1.899	1.911	...	1.924
PR(C)	0.862	0.739	0.742	0.746	0.748	...	0.750
PR(D)	0.296	0.275	0.270	0.276	0.277	...	0.277
PR(E)	1.332	1.381	1.401	1.411	1.417	...	1.423
PR(F)	1.189	1.256	1.277	1.288	1.293	...	1.300

2.4 Outils de recherche avancés

De nos jours, plusieurs outils de recherche avancés commerciaux ou publics sont inventés dans le but d'aider et d'assister les internautes lors de leurs recherches. L'outil de recherche A9[6], par exemple, analyse les activités et les préférences des internautes pour faire la recommandation des liens, des films ou des livres, etc. Le système Fooxx[7] lui aussi offre à ses utilisateurs la recommandation des pages web intéressant d'autres utilisateurs, il peut également mettre en contact plusieurs utilisateurs qui sont connectés en ligne, etc. Le principal atout de ces systèmes est qu'ils prennent les préférences et les activités (historiques) des internautes comme critère préliminaire pendant la recherche.
Nous présentons deux autres systèmes qui vont dans le même sens et qui sont issus du milieu académique.

2.4.1 CiteSeer

Bollacker *et al.* [15] ont réalisé un système qui utilise des moteurs de recherche pour trouver et télécharger des papiers (articles) ou des documents scientifiques potentiellement intéressants et reliés aux thèmes recherchés par des internautes.

[6] http ://www.a9.com
[7] http ://www.fooxx.com

Ces papiers sont analysés afin d'extraire certaines informations importantes (les caractéristiques sémantiques, les mots fréquents, les citations en référence, etc.), qui seront ensuite stockées dans une base de données. CiteSeer fonctionne de la même façon que les moteurs de recherche, en trois phases distinctes :

1. **L'acquisition des papiers** : cette étape consiste à exécuter un programme qui cherche à travers les moteurs de recherche (Altavista, HotBot, Excite, etc.) en combinaison avec des mots heuristiques comme "publications" et "postscript " les papiers et les documents scientifiques recherchés par les internautes. Il télécharge tous les fichiers ayant pour extension ".ps", ".ps.Z", ".ps.gz" et d'autres formats. C'est comme un "Crawler" dans les moteurs de recherche.

2. **L'analyse des papiers** : cette phase consiste à analyser les papiers téléchargés afin d'extraire toutes informations jugées nécessaires pour la phase de recherche, ensuite les stocker dans la base de données. Celle-ci va contenir, par exemple : les extraits des papiers, leurs URLs, les fréquences d'apparition des mots dans chaque extrait, les citations, etc. Ceci est similaire au programme d'indexation dans les moteurs de recherche.

3. **Recherche dans la base de données** : consiste à présenter à l'internaute les articles qui correspondent à sa recherche et les documents similaires. Pour mesurer la similarité entre plusieurs papiers (articles) CiteSeer utilise diverses méthodes. Il utilise la technique LikeIt [80] pour calculer la distance entre les entêtes, les institutions, auteurs et mots clés des papiers. Il utilise également la technique TFIDF (**T**erm **F**requency **I**nverse **D**ocument **F**requency) pour calculer le poids des mots des textes selon la formule suivante [15] :

$$w_{ds} = \frac{(0.5 + 0.5\frac{f_{ds}}{f_{dmax}})(log\frac{N_D}{n_s})}{\sqrt{\sum_{j \in d}((0.5 + 0.5\frac{f_{dj}}{f_{dmax}})^2(log\frac{N_D}{n_j})^2)}} \tag{2.2}$$

Où

w_{ds} : le poids du terme s

d : le document (papier)

f_{ds} : la fréquence du terme s

f_{dmax} : la fréquence maximale d'un terme dans tous les documents

f_j : la fréquence du terme j

N_D : le nombre de documents

n_s : le nombre de documents contenant le terme s

n_j : le nombre de documents contenant le terme j

Les mots *Stopped Words* comme "the", "as", "while", etc. sont ignorés par CiteSeer. Seules les racines des mots *Stems of Words* sont considérées, en utilisant la lemmatisation heuristique de Porter [64]. Par exemple, les mots "walking", "walk" et "walked" sont tous identiques au mot "walk".

Pour déterminer la similarité entre deux documents, CiteSeer calcule la distance entre leur vecteur associé contenant les poids des mots racines de chacun. Il y a plusieurs façons de calculer la similarité entre deux vecteurs. Soient X et Y deux vecteurs représentés respectivement par leurs vecteurs des poids $(p_{x1}, p_{x2}, p_{x3}, ..., p_{xn})$ et $(p_{y1}, p_{y2}, p_{y3}, ..., p_{yn})$. Les deux formules les plus souvent utilisées pour le calcul de similarité sont le **produit interne** et le **Cosinus** qui sont respectivement :

$$Sim(X, Y) = \sum_{i=1}^{n}(p_{xi}p_{yi}) \tag{2.3}$$

$$Sim(X,Y) = \frac{\sum_{i=1}^{n}(p_{xi}p_{yi})}{\sqrt{\sum_{i=1}^{n}(p_{xi})^2 \sum_{i=1}^{n}(p_{yi})^2}} \tag{2.4}$$

CiteSeer comprend la recherche de documents et la recherche de citations (voir figure 2.3). Il donne des statistiques de citations, des liens cités, des citations en contexte, des documents reliés, etc. De plus, il permet aux utilisateurs de faire des commentaires sur des articles.

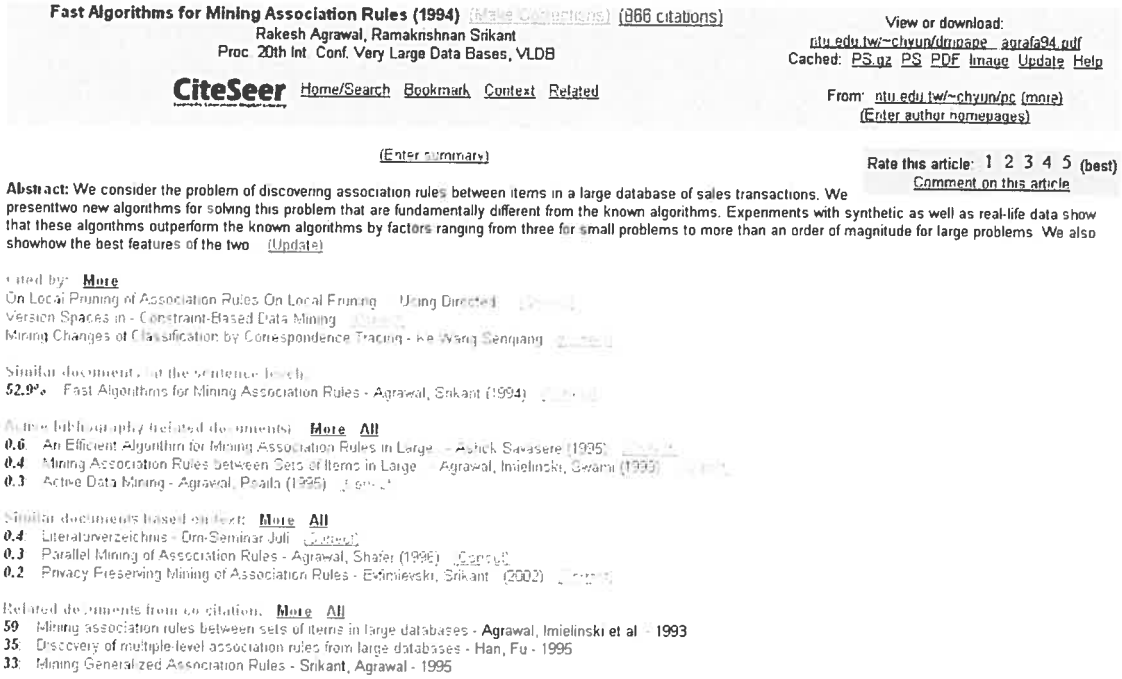

FIG. 2.3 – L'interface graphique de CiteSeer

2.4.2 I-SPY

Balfe et Smyth [9] ont conçu I-SPY, un méta-moteur de recherche utilisant une variété de moteurs de recherche (Google, HotBot, WiseNut, AllTheWeb, etc.). Son principal objectif est d'améliorer la pertinence des résultats des recherches. Quand un utilisateur envoie sa requête à I-SPY, ce dernier lui suggère des résultats reclassés selon les sélections effectuées auparavant sur ces résultats par d'autres

utilisateurs. I-SPY utilise une matrice appelée "Hit-Matrix", qui contient les recherches effectuées antérieurement par d'autres utilisateurs. Les lignes de "Hit-Matrix" correspondent aux requêtes formulées et les colonnes indiquent les pages web sélectionnées pour ces requêtes. À chaque fois qu'un utilisateur sélectionne la page web p_j pour la requête q_T, la valeur de la cellule H_{Tj} s'incrémente. Elle dénote le nombre de fois où la page web p_j a été sélectionnée pour la même requête q_T.

Par exemple, quand un utilisateur fait une nouvelle recherche sur la requête q_T, I-SPY l'intercepte et parcourt toutes les lignes de "Hit-Matrix" en vue de calculer la similarité entre cette nouvelle requête q_T à celles de "Hit-Matrix", selon la formule :

$$Sim(q_T, q) = \frac{|q_T \bigcap q|}{|q_T \bigcup q|} \tag{2.5}$$

Où

$Sim(q_T, q)$: la similarité entre la requête q_T et la requête q

Ensuite, I-SPY propose à cet utilisateur des résultats contenant des pages web pertinentes relativement à sa requête q_T. La pertinence d'une page web par rapport à une requête q_T est calculée selon la formule [9] :

$$Relevance(p_j, q_T) = \frac{H_{Tj}}{\sum_{i=0}^{n-1} H_{Ti}} \tag{2.6}$$

Où

$Relevance(p_j, q_T)$: la pertinence de la page p_j par rapport à la requête q_T.

H_{Tj} : le nombre de fois que la page web p_j a été sélectionnée pour la requête q_T.

H_{Ti} : le nombre de fois que la page web p_i a été sélectionnée pour la requête q_T.

n : le nombre de colonnes de la matrice "Hit-Matrix".

La figure suivante présente l'interface graphique du méta-moteur de recherche I-SPY.

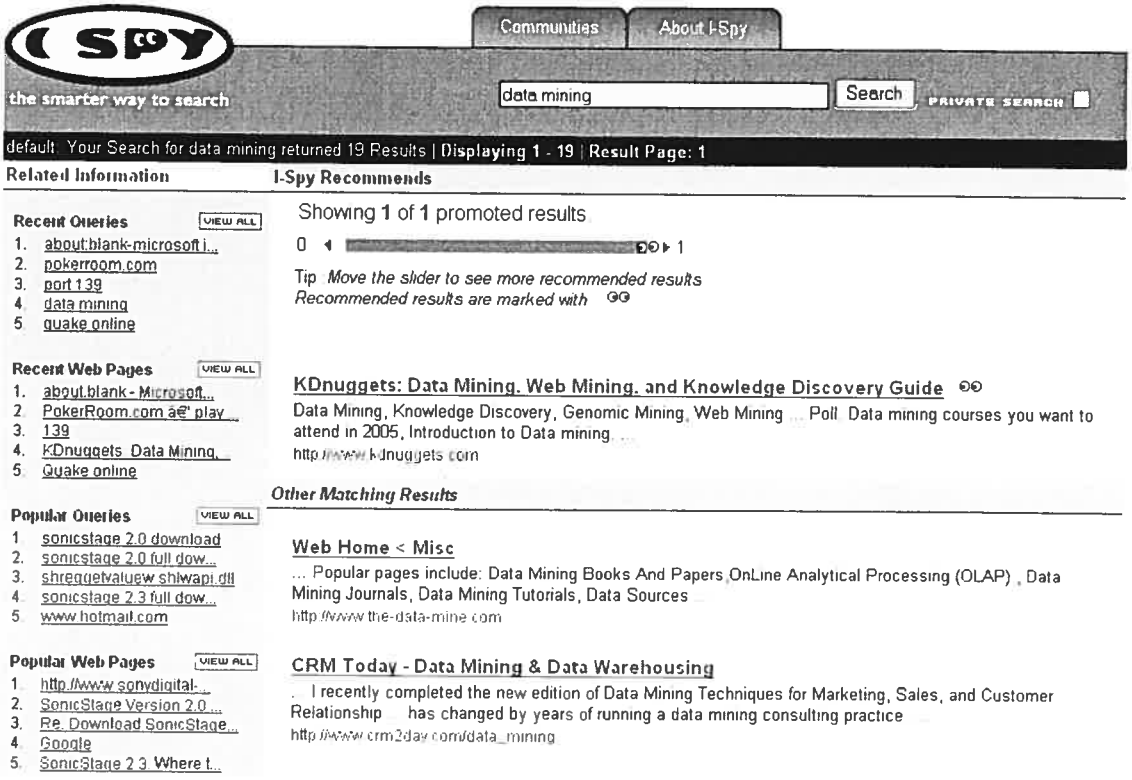

FIG. 2.4 – L'interface graphique d'I-SPY

2.4.3 Limitations de ces systèmes

L'idée d'assister les internautes pendant leurs recherches, en mettant des liens en avant ou en recommandant des liens susceptibles d'être intéressants, est un concept attrayant. L'implémentation de cette idée dans les systèmes I-SPY et CiteSeer fait bel et bien de ceux-ci des outils plus sophistiqués que des simples moteurs de recherche. Néanmoins, par exemple, les concepts proposés par le système I-SPY se basent effectivement sur les recherches précédentes de ses usagers, mais elles se focalisent essentiellement sur les requêtes des internautes. CiteSeer

quant à lui, considère à la fois les recherches de ses usagers et le contenu des documents.

L'idée proposée par I-SPY dans le but de reclasser (re-rank) les pages web est excellente. Pour une requête donnée, l'ordre de classement des pages web se fait selon le nombre de sélections qui ont été effectuées sur ces pages web par d'autres utilisateurs, sur des requêtes similaires à la requête en question. Toutefois, les requêtes des utilisateurs sont habituellement courtes, il faut alors combiner plusieurs techniques pour avoir une bonne similarité entre les requêtes.

CiteSeer quant à lui recommande les documents scientifiques en se basant fortement sur leurs contenus. Après les transformations nécessaires sur les documents (lemmatisation, extraction des vecteurs des poids des documents, etc.) CiteSeer calcule les similarités entre ces documents. Ainsi, en cherchant la requête "Association Rules" (voir figure 2.3) CiteSeer nous propose beaucoup d'informations supplémentaires qui pourraient être utiles, et lorsque nous cliquons sur le lien "Fast Algorithms for Mining Association rules (1994)" CiteSeer recommande d'autres articles similaires à celui-là tels que : les articles l'ayant cité, les articles en rapport à ce document basés sur le contenu, etc.

2.5 Conclusion

Ce chapitre a présenté une vue globale des outils de recherche standards et avancés, et a clarifié leur architecture générale et leur fonctionnement. Bien que le principal objectif des outils de recherche est d'aider l'utilisateur final à trouver, en un temps court, l'information adéquate qu'il recherche, le problème du volume de la toile (WWW) persiste toujours et s'impose continuellement, et ce malgré les nouvelles techniques employées pour affiner les résultats proposés à l'utilisateur.

Notre objectif est donc de concevoir un système capable d'assister, d'aider et de recommander des liens répondant aux besoins des utilisateurs. L'idée de base est d'appliquer la technique des règles d'association sur les documents qui sont jugés pertinents par un utilisateur ou un groupe d'utilisateurs. En effet, les règles d'association fournies par BLED lui permettront de prédire avec une certaine probabilité quels seront les documents pouvant intéresser davantage les utilisateurs, en se basant sur leurs documents préférés, contrairement aux autres systèmes qui font de la recommandation basée soit sur la similarité entre requêtes des utilisateurs (I-SPY), soit sur le contenu des documents (CiteSeer).

Nous allons définir dans le chapitre suivant le forage de données et détailler ses techniques et ses objectifs que nous utilisons pour réaliser notre système BLED.

Chapitre 3

Forage de données

Le présent chapitre est entièrement consacré au forage de données[1]. Il décrit ses tâches, ses objectifs, ses techniques ainsi que les étapes de son fonctionnement. Il détaillera entre autre la technique des règles d'association, car elle est utilisée par notre approche.

3.1 Introduction

Aujourd'hui, presque toutes les entreprises, petites ou grandes détiennent des sites web pour vendre leurs services. Des applications de bases de données gèrent leurs activités habituelles [65] (le support client, la gestion des produits, la comptabilité, le marketing, l'archivage, etc.). Toutefois, ces montagnes de données ne sont pas bien exploitées. Par exemple, voyons s'il est possible d'obtenir des réponses aux questions suivantes :

[1] Data Mining

- Comment établir une classification des prospects ?
- Comment classifier les clients par produits préférés ?
- Quels risques peuvent mettre en danger nos produits ?
- Quel est le profil des clients à long terme ?
- Comment attirer plus de clients ?
- Comment garder les clients pour toujours ?

Évidemment, si nous ne disposons pas de l'outil nécessaire, la réponse est non ! Ou avons-nous peut être de maigres chances d'avoir des réponses miraculeuses à ce scénario des questions. Ceci semble intéressant si nous voulons développer des stratégies claires et efficaces, pour pouvoir analyser et extraire des connaissances et les interpréter plus tard pour mieux connaître nos développements et nos clients. Pour répondre à tous ces besoins, la mise en place du forage de données devient alors incontournable.

3.2 Définitions

Le forage de données est :

1. Une activité complexe visant à extraire et synthétiser des informations inconnues, stockées dans un large volume de données [14].
2. Un outil puissant qui sert à trouver des informations cachées dans une base de données, dont le volume est important, et à donner une explication claire à ces informations [50].
3. Un processus décisionnel, où les utilisateurs cherchent des modèles d'interprétation de leurs données [33].
4. Une opération de tamisage d'un large volume de données afin de pouvoir dé-

couvrir de nouvelles corrélations, de nouvelles tendances, et de nouveaux modèles explicatifs des données [70].

5. Le forage de données se mesure à l'expression : *"Comment trouver un diamant dans un tas de charbon sans se salir les mains"* [33].

6. *"Un data warehouse, c'est comme la Californie en 1949, et la fouille de données la recherche de l'or. Sans la concentration d'or dans les rivières de Californie, les chances de succès des chercheurs d'or auraient été très limitées. Ainsi le data warehouse est un passage obligé pour le data mining"* [33].

Nous en déduisons que l'entrepôt de données permet de rendre disponibles les données en terme de quantité ou qualité pour toutes les opérations de forage de données telles que la prédiction, l'estimation et la segmentation.

Pour conclure, nous pouvons dire que le forage de données est un processus de sélection, d'exploration, de modification et de modélisation de grandes bases de données en vue de découvrir des relations entre ces données et fournir des explications claires à ces relations.

3.3 l'entrepôt de données

Bill Inmon a donné une définition très claire et complète au sujet de l'entrepôt de données[2] : *"Un data warehouse est une collection de données thématiques, intégrées, non volatiles et historisées, organisées pour la prise de décision"* [33, 65, 12, 47].

D'après cette définition, nous pouvons schématiser l'entrepôt de données comme une structure informatique, dans laquelle est stocké un volume important de don-

[2]data warehouse

nées afin que des personnes puissent accéder aisément à l'information dont elles ont besoin pour la prise de décision. Ces données contiennent des informations internes, des données de productions et des informations externes (voir figure 3.1). De plus, ces données sont :

1. **Organisées par thèmes** : les données sont organisées par sujets (consommateur, produit, ventes, etc.)
2. **Intégrées** : les données proviennent de sources multiples et hétérogènes (fichiers, bases de données externes, bases de données opérationnelles, transactionnelles, etc.).
3. **Historisées** : données d'archives créées au fil du temps.
4. **Non volatiles** : le stockage de données se fait indépendamment des bases de données opérationnelles et leur mise à jour est inhibée, c'est-à-dire, l'accès aux données est permis en lecture seule.

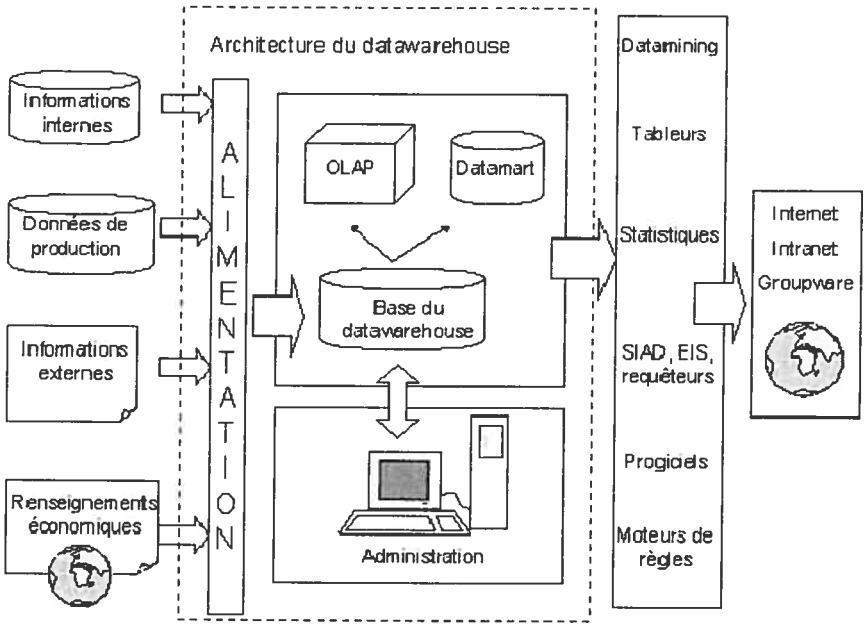

FIG. 3.1 – Principe d'un entrepôt de données

3.4 Magasin de données

Un magasin de données [3] [33] est un sous ensemble d'un entrepôt de données , qui ne contient que des données d'une activité bien déterminée de l'entreprise. Par exemple, le magasin de données de marketing ne contient que des informations propres au service marketing.

Afin de distinguer entre un entrepôt de données et un magasin de données, le tableau 3.1 ci-dessous illustre les principaux points qui les différencient :

TAB. 3.1 – Comparaison entre entrepôt de données et magasin de données [33]

	Entrepôt de données	Magasin de données
Cible utilisateur	Toute l'entreprise	Département
Implication du service informatique	Élevée	Faible ou moyenne
Modèles de données	A l'échelle de l'entreprise	Département
Champ applicatif	Multi sujets, neutre	Quelques sujets, spécifique
Sources de données	Multiples	Quelques unes
Stockage	Base de données	Plusieurs bases distribuées
Taille	Centaine de GO et plus	Une à 2 dizaines de GO

3.5 Tâches du forage de données

Nous avons dit que la fouille de données est une solution capable de mettre fin à certaines difficultés diverses, constatées quotidiennement dans un organisme. Quelque soit la nature du problème posé par le forage de données, la solution réside habituellement dans l'une des tâches suivantes : la classification[4], la des-

[3] Datamart

[4] Classification

cription[5], l'estimation[6], le groupement par affinité[7], la prédiction[8] et la segmentation[9][14].

3.5.1 La classification

La classification consiste à examiner les caractéristiques d'un élément (objet) nouvellement présenté et l'affecter à une classe d'un ensemble déjà prédéfini. Cette tâche repose sur des techniques de forage de données comme les arbres de décision, le CBR (Case-Based Reasoning) et éventuellement l'analyse des liens (voir section 4.2.3.2).

Exemples

- Évaluer des demandes de crédit ;
- Détecter les tendances boursières ;
- Examiner les demandes de remboursement ;
- Diagnostiquer des maladies.

3.5.2 La description

La description permet de décrire les données d'une base de données très complexe en vue d'en fournir des explications. Les techniques de forage de données opérantes pour cette tâche sont les arbres de décision et les statistiques en général.

---

[5]Description and profiling
[6]Estimation
[7]Association rules or affinity grouping
[8]Prediction
[9]Clustering

Exemples

- Les clients ayant acheté des films en DVDs classés "**action**" sont des hommes, célibataires et âgés de 20 à 35 ans.

3.5.3 L'estimation

L'estimation consiste à estimer une **variable continue** en fonction des variables dites explicatives, qualitatives ou quantitatives. Par exemple, pour le cas de la régression linéaire, l'estimation d'une variable Y en fonction des variables : $X_1, X_2, \cdots, X_n$ peut être traduite mathématiquement par : $Y = a_0 + a_1X_1 + a_2X_2 + \ldots + a_nX_n$.

Où

$a_0, a_1, \cdots a_n$ sont des constantes.

$X_0, X_1, \cdots X_n$ sont des variables dites explicatives.

Les techniques les plus appropriées à l'estimation sont : la régression, le CBR et les arbres de décision.

Exemples

- Estimer les revenus d'un client ;
- Estimer les bénéfices d'une compagne publicitaire.

3.5.4 La prédiction

Cette tâche est similaire à celles de la classification et de l'estimation, sauf qu'elle a pour objectif de prédire les comportements des objets ou d'estimer les valeurs futures. Les techniques opérantes pour cette tâche sont : les arbres de décision, les réseaux de neurones, la régression, les règles d'association et le rai-

sonnement à base de cas.

Exemples

- Prédire le salaire qu'une personne peut espérer ;
- Prédire le départ d'un client ;
- Prédire la durée d'hospitalisation d'un patient.

3.5.5 La segmentation

La segmentation consiste à créer des groupes homogènes, qui se ressemblent le plus à partir de données hétérogènes. Les groupes homogènes seront interprétés par un expert du domaine qui déterminera leur signification et l'intérêt de chaque groupe ainsi obtenu. Grâce à cette tâche, il est possible d'attribuer des groupes préétablis pour servir la tâche de la classification, de l'estimation et de la prédiction. La technique la plus appropriée à la segmentation est l'analyse des clusters.

Exemples

La segmentation peut réaliser des opérations comme :

- Segmenter la clientèle pour une campagne promotionnelle ;
- Détecter les groupes aberrants : fraudes, intrusions, etc.

3.5.6 Le groupement par affinité

Cette tâche consiste à identifier les dépendances qui existent entre les caractéristiques observées sur un ensemble de données. Elle est connue sous le nom de **"l'analyse du panier de la ménagère"**, car elle permet de connaître les produits qui vont naturellement ensemble dans un supermarché. Il est à noter que notre travail se focalise grandement sur cette tâche, c'est-à-dire, de trouver

les services/ documents sur Internet qui pourront être consultés ensemble pour ensuite en faire la recommandation aux autres internautes.

Parmi les techniques les plus appropriées à cette tâche, nous citons la technique des règles d'association, dont nous verrons les détails plus loin. Le groupement par affinité peut servir à d'autres tâches.

Exemples

- Analyser le choix des cours des étudiants ;
- Identifier des occasions de ventes croisées[10] ;
- Concevoir des groupements attrayants de produit ;
- Joindre les références d'un individu entre elles.

3.6 Étapes du processus de forage de données

Dans cette section, nous décrivons les différentes étapes du processus de fouille de données illustrées à la figure 3.2. Il est clair que le but final de la fouille de données est d'analyser des données afin d'en dégager des connaissances qui y sont masquées. Ce processus doit être d'abord précédé par l'identification des besoins de l'entreprise et l'identification de toutes les actions à entreprendre au cours de cette opération.

Selon la majorité des spécialistes en forage de données, le processus de forage de données suit les étapes suivantes :

1. Préparation des données[11] ;
2. Nettoyage des données[12] ;
3. Intégration des données[13] ;

[10] Cross-sale
[11] Data processing
[12] Data cleaning
[13] Data integration

4. Normalisation et transformation des données[14] ;

5. Recherche des modèles[15] ;

6. Évaluation et interprétation[16].

Il est à noter qu'il est possible de faire un retour arrière à n'importe quelle étape du processus de data mining, si le besoin s'en fait sentir.

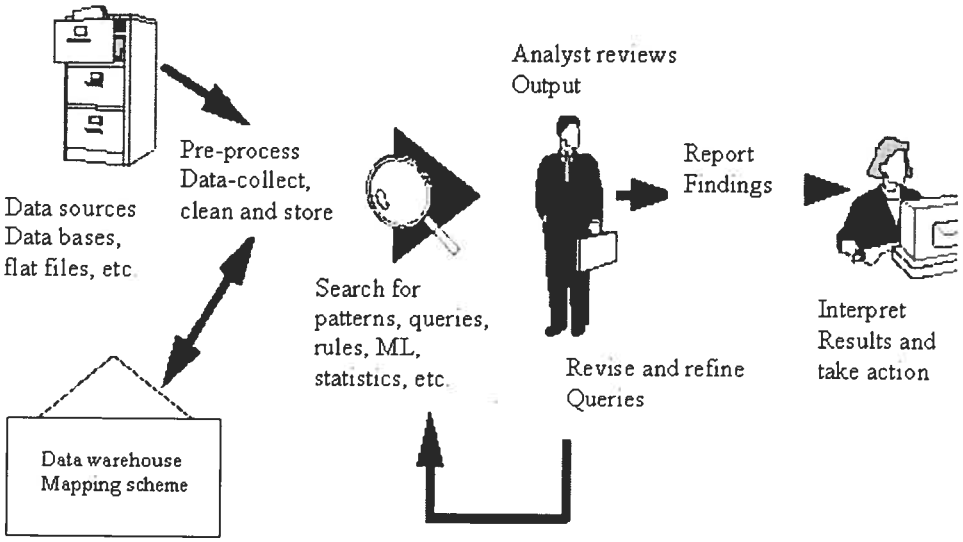

FIG. 3.2 – Processus de forage de données

3.6.1 Préparation des données

Cette étape consiste à réunir toutes les données nécessaires qui contribueront à la réalisation des objectifs imposés pendant la phase d'identification des besoins de l'entreprise. Ces données proviennent généralement de l'entrepôt de données, du magasin de données, des fichiers binaires, des fichiers textes ou même des bases de données opérationnelles (voir figure 3.2). A ce stade, certaines transformations

[14]Data normalization and transformation
[15]Search for patterns, queries, rules, etc.
[16]Interpretation of results

devraient se faire sur ces données afin d'éliminer toutes variables (attributs) inutiles pour l'opération de forage de données d'une part, et de retrouver celles qui s'ajustent aux objectifs imposés à priori. Il est à noter que ces transformations sont menées par des outils spécifiques comme : SQL et OLAP, qui nécessitent évidement des connaissances de données présentes dans les sources de l'entreprise.

3.6.2 Nettoyage des données

Une fois toutes les variables nécessaires sont identifiées, il est recommandé de nettoyer certaines données erronées. En effet, il arrive parfois que des variables (champs d'un formulaire) qui doivent être remplies à la main, par des agents de saisie n'ont pas été contrôlées par le programme, ceci provoque souvent diverses erreurs. Là encore, des outils comme SQL et OLAP sont incontournables, notamment pour examiner les redondances et l'intégrité des données.

Exemples

- Correction des erreurs de saisie ;
- Elimination des redondances des données ;
- Vérification de l'intégrité des données ;
- Mise à jour des informations incomplètes.

3.6.3 Intégration des données

Il est parfois important de se procurer des bases de données externes, si cela est nécessaire bien sûr, comme par exemple, une base de données démographique ou géographique. L'objectif est donc de consolider et d'enrichir les bases de données sur lesquelles le processus de forage de données va s'executer. Dans notre projet,

nous avons intégré la base de données des adresses IP mondiales afin d'élaborer des statistiques sur les recherches des internautes par pays.

3.6.4 Normalisation et transformation des données

Cette étape est très délicate, car elle est complètement dépendante de l'étape qui la suit, c'est-à-dire, le choix des techniques de forage de données. La normalisation et la transformation de données consiste à faire des transformations dans la base de données, parmi les opérations les plus connues, figure la discrétisation, qui consiste à transformer les variables continues (quantitatives) en variables qualitatives convenables. Par exemple, de transformer une variable de type date en variable de type numérique ; de créer de nouvelles variables supplémentaires dans la base de données (Moyenne, Max, Min, etc.) ; de transformer des variables nominales en variables numériques, ainsi la séquence : Excellent, Très bien, Bien, Moyen et Faible deviendra : 5, 4, 3, 2 et 1. Ces transformations doivent tenir compte des techniques de forage de données qui vont s'opérer sur la base de données, car certaines méthodes de forage de données comme les réseaux de neurones ou les statistiques classiques ne traitent que des données entièrement numériques.

3.6.5 Recherche des modèles

Le choix des techniques représente l'étape la plus importante du processus de forage de données. Ces techniques sont divisées en deux grandes classes [38] : a) *les techniques prédictives* permettant de généraliser de nouvelles informations en se basant sur des informations présentes, c'est-à-dire, qu'il y ait toujours des variables cibles à prédire ; b) *les techniques descriptives ou exploratoires* permettant à mettre en évidence des informations cachées dans un volume extraordinaire de

données, c'est-à-dire, qu'il n'y a pas de variables à prédire. Puis, le choix des techniques se fera selon les objectifs visés par l'étude (prédictifs ou descriptifs). Par exemple, si nous voulons faire de la classification nous pourrons utiliser les méthodes prédictives telles que : les arbres de décision ou raisonnement à base de cas.

Parmi les techniques de forage de données les plus populaires, nous trouvons les règles d'association, les réseaux de neurones, les arbres de décision, les réseaux bayésiens et le raisonnement à base de cas. Il est à noter que ces techniques permettent de produire des *Modèles* [17] représentants : des segments, des règles d'associations, des relations et des procédures de classifications, etc.

3.6.6 Évaluation et interprétation

Cette étape représente la phase finale du processus de forage de données. Elle consiste tout simplement à interpréter et évaluer [71] les modèles obtenus aux étapes précédentes. Parmi les critères que nous devons considérer, nous citons : la fiabilité, la robustesse, le temps de réponse et la facilité de compréhension. Toutefois, pour valider un modèle, il est judicieux de mesurer les impacts des actions engagées auparavant.

3.7 Techniques de forage de données

Nous avons mentionné au paragraphe précédent que les techniques de forage de données constituent le cœur de processus du forage de données. En effet, ces techniques permettent d'élaborer des modèles pour l'aide à la décision à partir

[17] Pattern

de données brutes.

Cette section décrira les techniques de forage de données les plus employées par la majorité des logiciels récents. Ces techniques sont basées sur des aspects mathématiques et statistiques très complexes, elles sont aussi complémentaires aux outils classiques utilisés dans le temps comme : les statistiques élémentaires basées sur la moyenne, les écarts types, les variances, SQL et Excel. D'autres outils de visualisation de données en histogrammes, les nuages de points, les graphes de contingence et les données tridimensionnelles. Parmi ces techniques nous trouvons :

1. Les arbres de décision[18] ;
2. Les réseaux bayésiens[19] ;
3. Les réseaux de neurones[20] ;
4. Le raisonnement à base de cas[21] ;
5. L'analyse des clusters[22] ;
6. Les règles d'association[23].

3.7.1 Arbres de décision

Les arbres de décision sont parmi les techniques les plus utilisées en forage de données, notamment dans la tâche de classification. Comme son nom l'indique, un arbre de décision est un arbre inversé contenant une racine, des nœuds, des arcs et des feuilles. Les nœuds internes de l'arbre contiennent les attributs sur lesquels des tests peuvent être effectués. Par convention, l'arc gauche correspond

[18] Decision tree
[19] Bayesian network
[20] Neural network
[21] Case base reasoning
[22] Cluster analysis
[23] Association rules

à une réponse positive de test. Les feuilles de l'arbre sont des nœuds qui n'ont pas d'enfants et contiennent les données réparties en classes identiques. La procédure de classification obtenue se traduit en règles de décision faciles à comprendre et applicables à d'autres nouvelles données. Les principaux algorithmes connus sont C4.5, CART, CHAID, FACT, ID3 [35, 44].

3.7.2 Réseaux Bayésiens

Les réseaux bayésiens sont inspirés quant à eux du fameux théorème de Bayes [35, 44], qui permet de calculer la probabilité postérieure d'un événement ou une hypothèse inconnue, sachant que d'autres événements ont été observés.
La formule de Bayes se présente comme suit :

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \tag{3.1}$$

- $P(H|X)$: Exprime la probabilité que l'hypothèse H soit vraie sachant qu'un autre événement X a été produit ou observé
- $P(X|H)$: Désigne la probabilité que l'événement X soit observé sachant que l'hypothèse H est vraie.
- $P(H)$: Représente la probabilité que l'hypothèse H soit vraie indépendamment de la valeur de X.
- $P(X)$: Indique la probabilité que l'événement X soit observé.

Les réseaux bayésiens sont très pratiques en forage de données, et particulièrement pour servir la tâche de la prédiction et de l'estimation. En effet, les réseaux bayésiens permettent de trouver des modèles de décision, qui permettent de prédire avec probabilité que certaines données appartiennent ou pas à une certaine catégorie.

3.7.3 Réseaux de neurones

Les réseaux de neurones [35, 44] ont été proposés en 1943 pour la première fois par McCulloch et Pitts, neurologues de l'université de Chicago, dans le but de créer un modèle mathématique du cerveau humain baptisé neurone formel. En 1949 un mécanisme d'apprentissage a été proposé par Donald Hebb sous forme d'une règle de modification des connexions synaptiques. Une décennie plus tard, Rosenblatt avait conçu le perceptron, un réseau de neurones artificiels capable d'apprendre, d'identifier des formes et de faire certains calculs.

En résumé, un réseau de neurone est formé de plusieurs unités de base appelées neurones formels, lequel calcule la somme pondérée ($\sum P_i E_i$) de ses entrées (voir figure 3.3 dénotées par $X_1, X_2, \cdots, X_n$). Cette somme sera injectés à une fonction de transfert qui calculera à son tour la sortie du réseau de neurone S. Le seuil α représente le taux d'erreur entre la valeur réelle C et la valeur S calculée par le réseau de neurone.

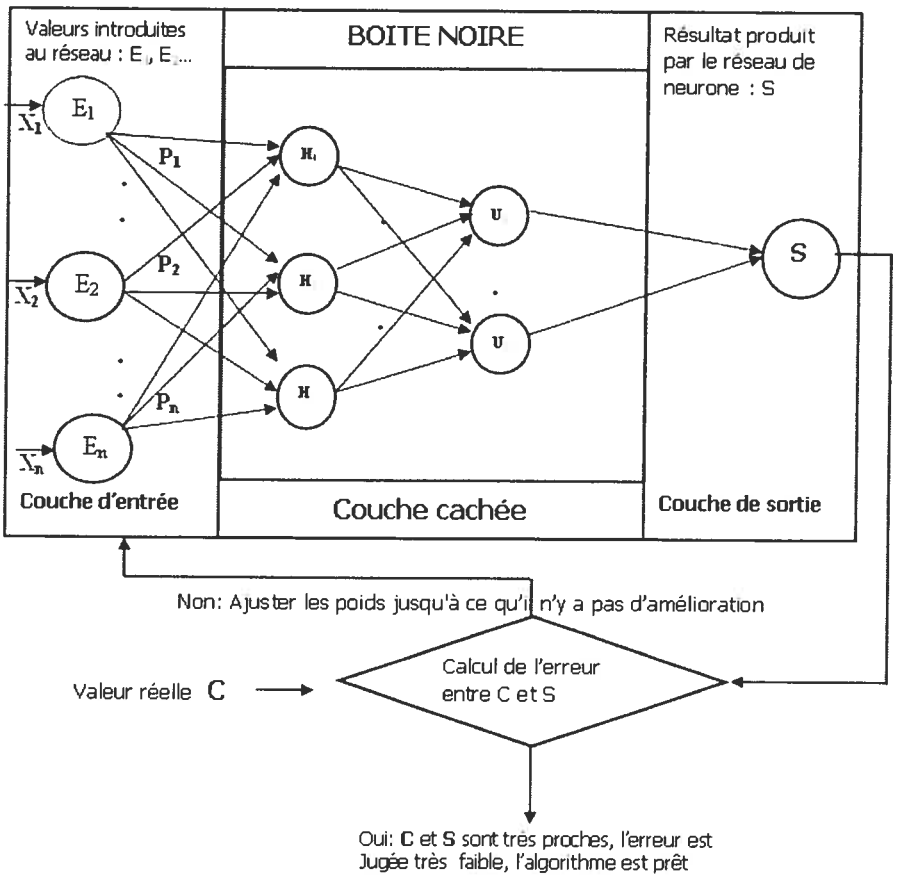

FIG. 3.3 – Réseau de neurones

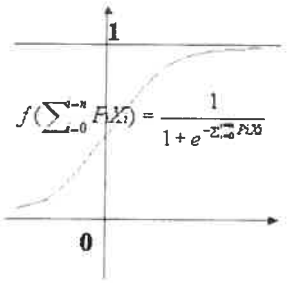

FIG. 3.4 – Fonction Sigmoïde

3.7.4 Raisonnement à Base de Cas

Le raisonnement à base de cas est une technique dans laquelle l'expérience antérieure peut s'appliquer à de nouvelles situations. Autrement dit, nous nous basons sur des cas résolus dans le passé pour trouver des solutions à des problèmes similaires actuellement affrontés. Si l'un des nouveaux problèmes n'a aucune référence dans la base de cas comprenant les experiences passées, il sera interprété, indexé et mis à jour dans cette base. Le raisonnement à base de cas est très vital surtout dans les activités où le rôle de l'expérience est décisif, c'est-à-dire, les activités où nous ne disposons ni de théories, ni de modèles formels qui les résolvent. En forage de données, le raisonnement à base de cas est très pratique pour les tâches de : la prédiction, l'estimation et notamment la classification, où il s'agit de positionner des objets nouvellement présentés par rapport aux plus proches voisins déjà résolus [1].

3.7.5 L'analyse des clusters

Les techniques de l'analyse des clusters [13, 46, 44] sont très utilisées en forage de données, car elles permettent de regrouper les individus d'une base de données en groupes disjoints, selon des principes de la similarité qui y existent entre individus. Comme nous l'avons déjà expliqué à la section 3.5, la segmentation sert à transformer les données d'une base complexe en une base plus compréhensible. Plusieurs techniques existent, nous en proposons quelque-unes :
K-Mean [38], K-medoids [38], Agglomération [38], CLARANS (A Clustering Algorithm based on Randomized Search) [38], BIRCH (Balanced Iterative Reducing and Clustering Hierarchies)[38], CURE (Cluster Using REpresentatives)[38], DBSCAN (Density-Based Clustering Method Based On Connected Regions with

Sufficiently High Density) [38], OPTICS [38] (Ordering Points To Identify the Clustering Structure), DENCLUE (DENsity-based CLUstering)[38], STING (STatistical INformation Grid)[38], WaveCluster [38] et CLIQUE (Clustering In QUEst) [38]. Ces algorithmes reposent sur certaines formules de calcul de distance :

1. **La distance Euclidienne :** $d_{Eucl}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
2. **La distance Manhattan :** $d_{Manh}(x,y) = \sum_{i=1}^{n} |x_i - y_i|$
3. **La distance de Minkowski :** $d_{Mink}(x,y) = \sqrt[q]{\sum_{i=1}^{n}(x_i - y_i)^q}$ où $q \in \mathbb{N}$
4. **La moyenne :** $Moyenne(x,y) = (\frac{\sum_{i=1}^{n} X_i}{n}, \frac{\sum_{i=1}^{n} Y_i}{n})$

où $\overrightarrow{x}(x_1, x_2, \ldots, x_n)$ et $\overrightarrow{y}(y_1, y_2, \ldots, y_n)$ sont deux objets de $\mathbb{R}$

3.7.6 Règles d'association

La technique des règles d'association [3, 2] a été inventée pour analyser des données transactionnelles des consommateurs des supermarchés. L'objectif était alors de découvrir toutes les affinités entre produits achetés par les clients afin d'implémenter de nouvelles démarches envers les clients et des produits. D'ailleurs, l'exemple type des règles d'association est l'analyse du ticket de caisse, dite "*L'analyse du panier de la ménagère*"[24].

Aujourd'hui, cette technique s'étend vers d'autres secteurs d'activités, où nous pouvons faire le regroupement des produits, des services ou des événements tels que : le commerce électronique, la santé, les télécommunications, la météo, etc. L'exemple suivant permet de mieux comprendre la technique des règles d'association.

Dans un supermarché, l'analyse du ticket de caisse a donné le tableau suivant :

[24] Market Basket Analysis

TAB. 3.2 – Liste d'achats

Transaction	Articles (items)
T00	A, B, C, D, E
T01	A, B, C
T02	A, C, D, E
T03	C, D, E
T04	A, E
T05	D, E
T06	A D, E
T07	E
T08	B, C
T09	C, D

Chaque ligne de ce tableau contient une transaction (un achat) identifiée par un numéro unique et un ensemble d'articles qui lui sont associés. Par exemple, dans la ligne 1, il s'agit d'un client $T00$, qui a acheté les produits A, B, C, D et E. La technique des règles d'association permet d'identifier les associations entre les articles de cette liste d'achats et produit des règles d'association comme celle-ci :

- 60% des achats contenant l'item **A** contiennent également l'item **C**, et **30%** de cette liste d'achats ne contiennent que les items **A** et **C**

Dans cette règle, le rapport **60%** indique la proportion des clients ayant acheté l'article **C** parmi ceux qui ont acheté l'article **A**. Alors, que le taux **30%** représente la proportion où les articles **A** et **C** sont présents ensembles dans la liste d'achats. Ces deux seuils, c'est-à-dire, **60%** et **30%** représentent en vérité les valeurs clés de cette règle d'association, ils sont appelés respectivement "**confiance**" et "**support**" de la règle, et ils déterminent l'importance et la signification de cette règle.

3.7.6.1 Découverte des itemsets fréquents

Soient $\mathcal{I} = \{i_1, i_2, i_3, \cdots, i_m\}$ un ensemble de m items et $\mathcal{B} = \{t_1, t_2, t_3, \cdots, t_n\}$ une base de données de n transactions. Chaque transaction t_i est identifiée par un TID (Transaction IDentification), elle est constituée d'un sous ensemble $I \subseteq \mathcal{I}$ de taille k (items) appelé $k - itemset$ [4, 2].

1. **Un support d'un itemset** I est le pourcentage des transactions de $\mathcal{B}$ qui contiennent l'itemset I [4, 2] :

$$Support(I) = \frac{|t \in \mathcal{B}/I \subseteq t|}{|t \in \mathcal{B}|} \tag{3.2}$$

Exemple

Dans la liste d'achats du tableau 3.2.

$$support(A) = \frac{5}{10} = 50\%$$

$$support(B) = \frac{3}{10} = 30\%$$

$$support(A, E) = \frac{4}{10} = 40\%$$

2. **Un itemset fréquent** est un itemset dont le support est supérieur ou égal au seuil minimal appelé *minsupport* défini par l'utilisateur [4, 2].

3. **Une Règle d'association** est une implication de la forme $I_1 \Longrightarrow I_2$ entre deux itemsets I_1, I_2 telle que $I_1 \bigcap I_2 = \emptyset$. Cette règle d'association possède un support s et une confiance c définis comme suit :

$$\{R : I_1 \Longrightarrow I_2, Support(I_1 \bigcup I_2) = \frac{|\mathcal{T} \in \mathcal{B}/I_1 \bigcup I_2 \subseteq \mathcal{T}|}{|\mathcal{B}|}, Confiance = \frac{Support(I_1 \bigcup I_2)}{Support(I_1)}\}$$

4. **Le support d'une règle d'association** est la fréquence d'apparition simultanée des items qui apparaissent dans la condition (prémisse) et la conséquence (résultat) de la règle, dans la base des transactions $\mathcal{T}$ de $\mathcal{B}$ [4, 2].

Exemple

Considérons la règle d'association $\{\mathrm{R} : A \longrightarrow C\}$ déduite du tableau 3.2.

$$support(R) = \frac{3}{10} = 30\%$$

5. **La confiance d'une règle d'association** est le rapport entre le nombre de transactions contenant tous les items figurants dans la règle et le nombre de transactions contenant seulement les items de la condition (prémisse) de la règle, dans la base des transactions $\mathcal{T}$ de $\mathcal{B}$[4, 2].

Exemple

Considérons la règle d'association $\{\mathrm{R} : A \longrightarrow C\}$ déduite du tableau 3.2.

$$Confiance(R) = \frac{3}{5} = 60\%$$

3.7.7 L'algorithme APRIORI

L'extraction des règles d'association [3, 2] consiste à déterminer toutes les règles d'association pour lesquelles le support et la confiance sont supérieurs ou égaux respectivement à un seuil de support minimum appelé *minsupport* et à un seuil de confiance minimum appelé $minconfiance$ fixés selon les objectifs et la nature des données à traiter.

La première opération de l'extraction des règles d'association consiste à compter les occurrences de chaque item de l'ensemble $\mathcal{I}$ afin de déterminer les $1-itemsets$ fréquents. Alors que, la k^{me} itération permet de créer les ensembles candidats C_k

par la procédure **Apriori-Gen** [4, 2] (voir **Algorithm 2**) en utilisant les F_{k-1} de $k-itemsets$ fréquents créés à la $(k-1)^{ime}$ itération. Puis, nous calculons le support de chaque candidat de C_k afin de ne garder que les candidats de C_k qui sont contenus dans une transaction t donnée avec un support supérieur ou égal au *minsupport* ,c'est-à-dire, les F_k de $k-itemsets$ fréquents. La procédure **Subset**(C_k, t) [4, 2] reçoit en entrée l'ensemble C_k de $(k-1)itemsets$ candidats et l'objet t de l'ensemble $\mathcal{B}$ et fournira en sortie l'ensemble C_t candidats contenant l'objet t.

Algorithm 1 Extraction des itemsets avec APRIORI

Entrée: Base de données transactionnelles $\mathcal{B}$; seuil minimal de support *minsupport*;
Sortie: $\bigcup_k F_k$ des k-itemsets fréquents;

Début
pour $(k := 2;\quad F_{k-1} \neq \emptyset;\quad k++)$ **faire**
  $C_k := Apriori-Gen(F_{k-1})$;// génération des candidats.
  pour tout transaction $t \in \mathcal{B}$ **faire**
    $C_t := Subset(C_k, t)$; // candidats contenus dans t
    pour tout candidat $c \in C_t$ **faire**
      $c.support++$;
    fin pour
    $F_k = \{\quad c \in C_k \quad | \quad c.support \geqslant \quad minsupport\}$;
  fin pour
fin pourRetourner $\bigcup_k F_k$;
Fin

3.7.8 La procédure Aprioi-Gen

Cette procédure [4, 2] reçoit en entrée l'ensemble F_{k-1} de $(k-1)-itemsets$ fréquents et fournira en sortie l'ensemble C_k de $k-itemsets$ candidats, notons que la génération des $k-itemsets$ est un problème exponentiel. **Apriori-Gen()** établit une jointure entre deux itemsets fréquents p et q de taille $k-1$ de l'ensemble F_{k-1}. Le résultat de cette opération est inséré dans l'ensemble C_k si les

$k-2$ premiers items qui les composent sont égaux. En outre, **Apriori-Gen()** efface tous les $(k-1)-itemsets$ candidats dont l'un des sous-ensembles s de taille $k-1$ ne figure pas dans l'ensemble précédent F_{k-1} (voir l'algorithme ci-dessous).

Algorithm 2 Génération des itemsets candidats avec Apriori-Gen

Entrée: ensemble F_{k-1} de (k-1)-itemsets fréquents ;
Sortie: ensemble C_k de k-itemsets candidats ;

Début
insert into C_k ;
Select $p.item_1, p.item_2, p.item_3, p.item_4, \cdots, p.item_{k-1}, q.item_{k-1}$;
from F_{k-1} p, F_{k-1} q ;
where $p.item_1 = q.item_1, \cdots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$;
pour tout itemsets candidat $c \in C_k$ **faire**
  pour tout sous-ensemble s de c de taille $k-1$ **faire**
    si $s \notin F_{k-1}$ **alors**
      supprimer c **de** C_k ;
    fin si
  fin pour
fin pour
Retourner C_k ;
Fin

Afin de mieux comprendre le fonctionnement des deux algorithmes présentés précédemment, prenons l'exemple de la **figure 3.5**, nous avons une liste d'achats $\mathcal{D}$ contenant 9 transactions, dont chacune est identifiée par un TID et un sous ensemble d'articles quelconques. Nous notons qu'un article peut désigner des objets différents, dans un contexte bien déterminé. Par exemple, dans le cas du supermarché, les objets peuvent être : café, lait, jus d'orange, etc. Dans notre contexte, un article peut désigner toute sorte de pages web (documents PDF, sons, images, etc.). Il faudrait ensuite fournir la valeur du seuil *minsupport* du paramètre d'entrée de l'algorithme APRIORI, en supposant que *minsupport* soit égal à 2/9.

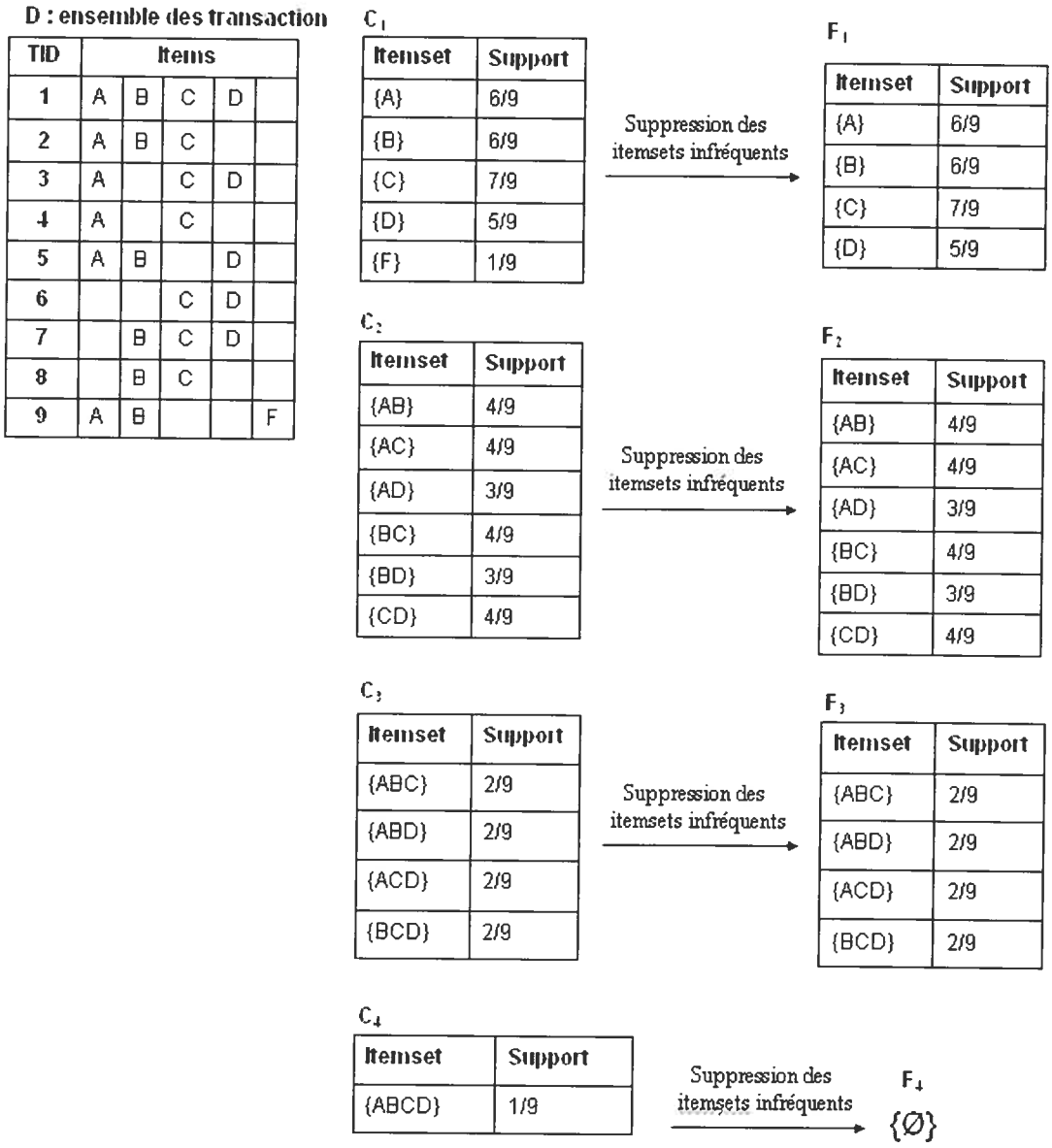

FIG. 3.5 – Extraction des itemsets fréquents par APRIORI

La première opération consiste alors à calculer les occurrences de chaque article pour obtenir les $1-itemsets$ fréquents de F_1=A, B, C, D. Nous remarquons que l'article F est élagué de l'ensemble C_1, car sa fréquence d'apparition dans la liste $\mathcal{D}$ est inférieure à 2/9 (voir figure 3.5 C_1). Nous réitérons ensuite l'algorithme APRIORI, qui fait appel de son côté à la procédure **Apriori-Gen** pour déterminer le reste des F_k de $k-itemsets$ tant que F_{k-1} n'est pas vide (voir figure 3.5 F_4). Après avoir terminé l'exécution de l'algorithme APRIORI, nous obtenons comme résultat final les ensembles : F_1, F_2 et F_3 à partir desquels les règles d'association seront générées.

3.7.9 Génération des règles d'association

Nous avons vu qu'une règle d'association n'est qu'une relation entre les itemsets de l'ensemble F des itemsets fréquents, celle-ci se présente sous la forme : $\{R : I_1 \Longrightarrow I_2\}$, dans laquelle I_1 et I_2 sont des itemsets fréquents appelés respectivement l'antécédent (prémisse) et la conséquence de la règle R. Une règle est dite valide, si et seulement si le rapport $\frac{Support(I_1 \bigcup I_2)}{Support(I_1)}$ est supérieur ou égal au seuil $minconfiance$. Le processus de génération des règles d'association se déroule comme suit :

Pour chaque itemset fréquent I_1 de l'ensemble F de taille supérieure ou égale à deux, nous déterminons les sous ensembles I_2 de I_1, et la valeur du rapport $confiance = \frac{Support(I_1 \bigcup I_2)}{Support(I_1)}$. Une règle $\{R : I_2 \Longrightarrow (I_1 - I_2)\}$ sera retenue si seulement si, le rapport calculé representant sa $confiance$ est supérieur ou égal au seuil de confiance $minconfiance$.

Algorithm 3 Génération des règles d'association avec Gen-Règles.

Entrée: ensemble F itemsets fréquents ; seuil minimal de confiance $minconfiance$;
Sortie: ensemble $\mathcal{AR}$ de règles d'association valides ;

Début
pour tout k-itemsets fréquents $l_k \in F$ tel que $k \geq 2$ **faire**
  $H_1 = \{1 - itemsetssous - ensemblesdel_k\}$;
  pour tout $h_1 \in H_1$ **faire**
    $confiance(r) = support(l_k)/support(l_k - h_1)$;
    si $(confiance(r) \geq minconfiance)$ **alors**
      $\mathcal{AR} = \mathcal{AR} \bigcup \{r : (l_k - h_1) \longleftarrow h_1\}$;
    sinon
      $H_1 = H_1 \backslash \{h_1\}$;
    fin si
  fin pour
  Gen-Rules(l_k, H_1) ;
fin pour
Retourner$\mathcal{AR}$;
Fin

Algorithm 4 Insertion des règles d'association dans $\mathcal{AR}$ avec Gen-Règles

Entrée: k-itemsets fréquent l_k ; H_m de m itemsets conséquences ; $minconfiance$;
Sortie: ensemble $\mathcal{AR}$ de règles d'association valides ;

Début
si $k > m + 1$ **alors**
  $H_{m+1} = Apriori - Gen(H_m)$;
  pour tout $h_m \in H_{m+1}$ **faire**
    $confiance(r) = support(l_k)/support(l_k - h_{m+1})$;
    si $(confiance(r) \geq minconfiance)$ **alors**
      $\mathcal{AR} = \mathcal{AR} \bigcup \{r : (l_k - h_{m+1}) \longleftarrow h_{m+1}\}$;
    sinon
      supprimer h_{m+1} de H_{m+1} ;
    fin si
  fin pour
  Gen-Rules(l_k, H_{m+1}) ;
fin si
Fin

Exemple

Reprenons l'exemple défini précédemment (voir figure 3.5). L'ensemble F contient trois sous ensembles : F_1, F_2, F_3. La figure 3.6 montre la génération des règles d'association en déroulant les algorithme 3 et 4 sur l'ensemble F.

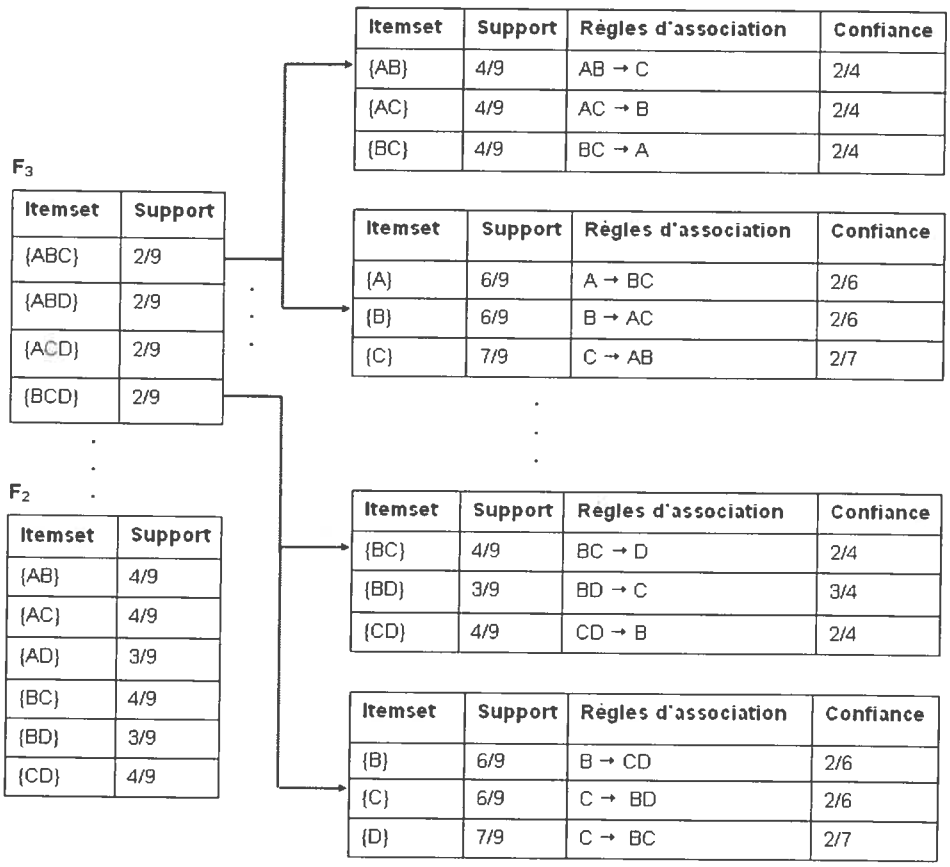

FIG. 3.6 – Exemple de génération des règles d'association

De nombreuses entreprises s'intéressent de plus en plus aux outils du forage de données. La figure 3.7 montre une liste non exhaustive de produits avec leurs descriptions.

Produit	Données	Techniques de modélisation
4Thought de Cognos	SGBD	**Réseaux de neurones**
Alice de Isoft	SGBD, Fichiers	**Arbres de décision**
Clementine de SPSS	SGBD, fichiers	**Multiples**
Datamind de D'Epiphany	SGBD, Fichiers	**Propriétaire (Proche de bayes)**
Intelligent Miner d'IBM	DB2, fichiers	**Multiples**
Knowledge Seeker de Angoss	SGBD, Fichiers	**Arbres de décision**
Predict de NeuralWare	SGBD	**Réseaux de neurones**
Previa de Elseware	Fichiers	**Réseaux de neurones**
SAS Entreprise Miner de SAS	SGBD et fichiers	**Multiples**
Saxon de Pmsi	Fichiers	**Réseaux de neurones**
SPAD de CISIA	SGBD, Fichiers	**Multiples**
Strada de Complex System	Fichiers	**Réseaux de neurones et algorithmes génétiques**
Wizwhy de WizSoft	SGBD, Fichiers	**Associations**

FIG. 3.7 – Quelques produits en forage de données

3.8 Conclusion

Nous avons exposé brièvement un domaine précieux dans la découverte de connaissances KDD (Knowledge Discovery in Databases), un outil puissant, efficace et capable de retrouver des informations cachées dans un large volume de données, avec lesquelles nous pouvons prédire et prévoir à court, moyen et long terme nos objectifs vis-à-vis de nos produits et nos clients.

Plusieurs outils sont disponibles, dont le choix se base sur plusieurs critères. Par

exemple, la multitude des techniques de forage de données implémentées par l'outil choisi, la qualité de ses algorithmes, les différents types de données qui peuvent être gérées par l'outil, la capacité de traitement de données volumineuses et la performance de l'outil.

Notre objectif de départ découle de l'implémentation de la technique des règles d'association du forage de données pour traiter des données provenant du web. C'est pourquoi, le chapitre suivant sera consacré à un domaine particulier de la découverte de connaissances appelé **forage du web**, qui consiste à employer de la fouille de données en vue d'extraire des connaissances à partir de données appelées **"données du web"**.

Chapitre 4

Forage du Web

Dans le présent chapitre, nous allons évoquer succinctement des notions se rapportant aux cookies, les fichiers journaux, le forage du web et sa taxonomie, nous présenterons ensuite, les données web suivies de la taxonomie de forage du web[1].

4.1 Généralités

Nous somme convaincus que l'objectif des chercheurs dans le domaine d'Internet, entre autre en commerce électronique, vise à collecter toutes informations jugées pertinentes sur les personnes ayant fréquenté leur site web, en examinant des ressources comme les fichiers journaux[2], les cookies [78] et les sessions [56, 45, 32, 73, 28]. Ces informations seront exploitées pour améliorer les perfor-

[1] Web Mining

[2] Log Files

mances de sites web, et analyser davantage les caractéristiques et les réactions des internautes pendant leur navigation sur le site. Parmi les informations les plus importantes, nous trouvons : Les pages web visitées, l'heure de visite, l'adresse IP de la machine appelante, le site à partir duquel l'utilisateur est arrivé, etc.

Le forage du web est l'une des branches les plus répandues dans le domaine du WEB (personnalisation des site web, systèmes de recommandation, extraction d'informations, etc.). Il consiste à employer des techniques et des algorithmes en vue de développer des approches et des outils permettant d'extraire des informations pertinentes cachées dans une gigantesque masse de données. Ces informations ont pour objectifs non seulement de cibler les internautes, mais également de prévenir et d'anticiper leurs attentes futures.

4.1.1 Les Cookies

Un cookie est un petit fichier texte envoyé à l'ordinateur de l'internaute par un serveur web, en passant par son fureteur lors d'une visite à un site web. Ce fichier contient des informations qui pourraient être réutilisées ou modifiées lorsque cet internaute se reconnectera ultérieurement sur le site web [78]. En effet, lorsqu'une adresse URL d'un site web est demandée par un internaute, son navigateur web commence d'abord par chercher et examiner tous les cookies présents dans l'ordinateur (disque dur) de cet internaute, s'il détecte ceux qui étaient affectés à cet URL il les réexpédie au serveur web qui les a créés. Par exemple, lorsqu'un internaute veut consulter son courrier électronique sur un service de messagerie, comme celui de Hotmail, il constatera certainement que son nom d'usager est déjà affiché sur sa page web d'accueil s'il a déjà autorisé son

fureteur à accepter les cookies, à ce moment, l'internaute n'a qu'à saisir son mot de passe pour accéder à sa boite de réception. En pratique, les cookies possèdent la structure suivante :

```
SetCookie :
Nom=VALEUR|Expires=DATE|Path=CHEMIN|Domain=DOMAINE|Secure
```

Tous ces attributs sont facultatifs, à part l'attribut `Nom`

TAB. 4.1 – Description d'un cookie

Attribut	Description
`Nom`	Suite de caractères, et de chiffres, servant à identifier la machine de l'utilisateur
`Expires`	Date d'expiration du cookie, une fois celle-ci atteinte, le cookie sera effacé du disque, ou invalidé dépendamment du type de fureteur du client. Le cookie expirera à la fin de la session si l'attribut "expires" n'a pas été mentionné lors de sa création
`Path`	Répertoire dans lequel le cookie est valide, les sous répertoires sont également acceptés.
`Domain`	Domaine auquel le cookie est affecté
`Secure`	Mode de transfert des pages web demandées par l'utilisateur. Il s'agit des connexions sécurisées.

En réalité, les cookies possèdent certaines limites et inconvénients :

1. Certains utilisateurs refusent systématiquement la réception des cookies ;
2. L'utilisateur peut effacer les cookies de son disque à n'importe quel moment ;
3. Par convention, la taille d'un cookie est limitée à 4 Ko ;
4. Le nombre maximum des cookies qui peuvent être stockés par navigateur est limité à 300 fichiers ;
5. Un serveur web ne peut envoyer plus de 20 cookies dans l'ordinateur de l'utilisateur. Ceci dit, le site le plus visité aura plus de chance de stocker plus de cookies ;

6- Un ordinateur peut être utilisé par plusieurs personnes, à l'université, à la

maison, au travail, etc. De même, un utilisateur peut utiliser plusieurs stations. Pour conclure, nous pouvons dire que les cookies en réalité permettent d'identifier la machine de l'internaute et non pas son identité.

4.1.2 Les fichiers journaux

Contrairement aux cookies, un fichier journal[3] est un fichier texte créé par un logiciel spécifique sur le serveur web et non pas sur la machine de l'internaute. Ce fichier sert à garder l'empreinte de toutes activités observées sur un site web vis-à-vis du monde extérieur [55]. L'analyse de tel fichier permet de mieux comprendre les progressions des utilisateurs sur un site web, d'optimiser sa structure (topologie) pour faciliter l'accès au contenu de ces pages web, d'évaluer son efficacité, de quantifier son succès et prévoir même de la vente croisée.
Les informations que contient un fichier journal sont des attributs, variant d'un standard à l'autre : W3C , NCSA , IIS , etc. [55]. Cependant, il y en a qui sont en communs comme :

1. L'adresse IP de la machine de laquelle se connectait l'utilisateur ;
2. Date et heure de connexion ;
3. Pages visitées ;
4. Taille du fichier téléchargé ;
5. Le site à partir duquel l'utilisateur est arrivé (site de renvoi).

Exemple

L'exemple suivant montre une entrée (enregistrement) d'un fichier journal.

Software : Microsoft Internet Information Services 6.0 Version : 1.0 Date : 2002-05-02 17 :42 :15 Fields : date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status cs(User-Agent) 2002-05-02 17 :42 :15 172.22.255.255 - 172.30.255.255

[3]Log file

80 GET /images/picture.jpg - 200 Mozilla

Dans cet exemple, nous remarquons que l'adresse IP de la machine est 172.30.255.255, la date et l'heure de la visite sont $2002-05-02$ et $17:42:15$, *picture.jpg* est le fichier visité.

Parfois, certains attributs ne sont pas identifiables, le serveur web les remplacera implicitement par le signe "-". Comme dans l'exemple précédent, l'attribut "username" a été remplacé par "-".

Comme pour les cookies, les fichiers journaux possèdent aussi des désavantages :

1. Un serveur web ne peut observer que l'adresse IP du serveur proxy requérant et non celles des utilisateurs cachés derrière lui. Un serveur proxy est un dispositif qui vise à conserver dans sa mémoire cache les données les plus habituellement demandées sur un réseau LAN (Local Area Network), afin de les rendre disponible en cas de besoin. Toutefois, cette stratégie s'est étendue à un autre point de mire, qui consiste à éluder les problèmes de saturation de la bande passante pour l'accès Internet à travers une seule adresse IP, et de conserver les pages visitées dans sa mémoire cache. La validité du contenu du cache dépend essentiellement de plusieurs paramètres tels que : *En-tête HTTP expire, en-tête HTTP Last-Modified*, la fréquence d'utilisation, etc. Ceci dit, les serveurs web ne peuvent jamais déceler l'adresse IP d'où les requêtes des internautes étaient émises ;
2. Un serveur proxy intercepte les requêtes provenant des utilisateurs, en cherchant d'abord dans son cache s'il y a des réponses correspondant à ces requêtes avant qu'elles soient redirigées vers le site web demandé. À ce moment, le serveur web perdra un tas d'informations relatives aux internautes qui pourraient être utiles [47] ;
3. Un navigateur web comprend une mémoire cache qui lui permet de stocker les pages web visitées par l'internaute. Ces pages web peuvent être consultées en passant par le bouton "*Précédent*". Ceci empêchera le serveur web d'acquérir plus

de données sur cet internaute ;

4. Enfin, lorsqu'un utilisateur ouvre une session connexion sur le réseau Internet. Son fournisseur d'accès Internet lui fournira une adresse IP dynamique, qui n'est pas forcement identique à celle attribuée auparavant. Ceci menace grandement l'identification des utilisateurs et la met en péril.

4.2 Forage du web

Le forage du web peut être défini, très naturellement, comme l'application des techniques de statistiques et de forage de données à une large masse de données appelée **données du web**[4], en vue de développer des outils qui permettront d'extraire des connaissances cachées dans cette masse de données [75].

4.2.1 Les données du web

Les données du web comprennent l'ensemble des données qui peuvent être utilisées en forage du web. La figure 4.1 montre ses principaux composants [72] : données du contenu[5], données de la structure[6], données d'usage[7] et données concernant le profil utilisateur[8].

[4]Web Data
[5]Content Data
[6]Structure Data
[7]Usage Data
[8]User Profile

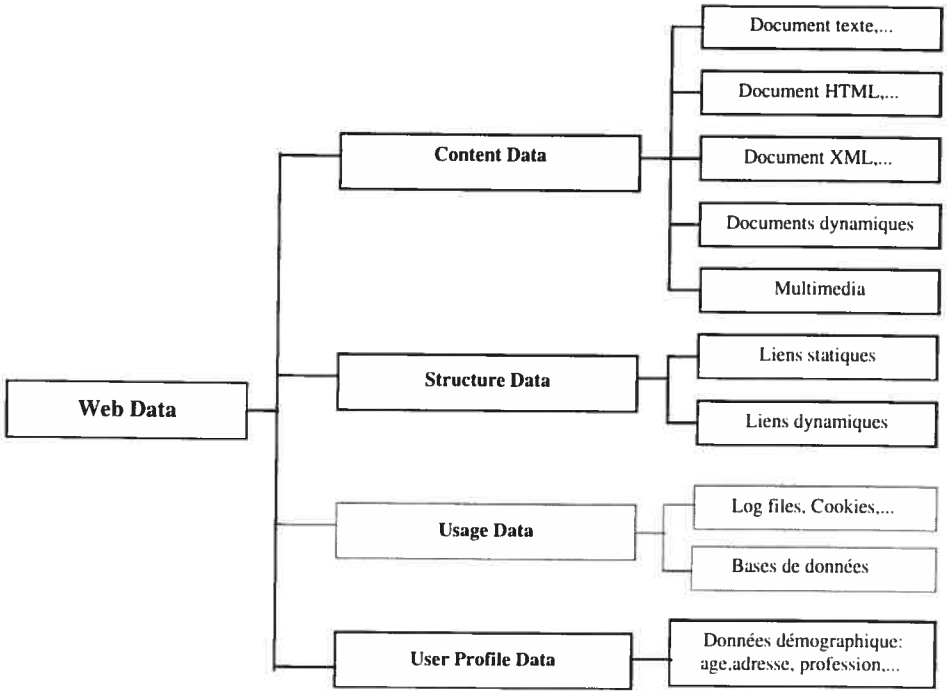

FIG. 4.1 – Les données web

4.2.1.1 Les données du contenu

Comme son nom l'indique, cette catégorie de données réunit toutes sortes de données qui peuvent être présentées aux internautes pendant leur navigation. Nous en distinguons trois types [72, 26] :

1. Le premier type intégrant des données appelées *données fortement structurées.* Nous entendons par cela, les données soumises à une structure bien déterminée lors de la conception. Elles sont stockées généralement dans des bases de données relationnelles ou objets ;

2. Le deuxième type renfermant des données appelées *données semi structurées.* Elles sont généralement représentées par des structures différentes en utilisant par exemple, le format XML, web sémantique, RDF, RDF-Schéma et OWL ;

3. Le dernier type comprend des données appelées *non structurées* incluant généralement des données formatées en HTML, texte, hypertextes, etc.

4.2.1.2 Les données de la structure

Cette catégorie de données retrace l'organisation interne des nœuds (pages web) d'un site web. Elle peut contenir des données brutes comme les tags : HTML, XML, RDF, etc., ou liens reliant les nœuds d'un site web ou liens raccourcissant l'accès à d'autres sites web [72, 26].

4.2.1.3 Les données de l'usage

Il s'agit des informations stockées dans les fichiers journaux qui décrivent toutes empreintes laissées par les utilisateurs du site web au cours de leurs visites. Ces informations se divisent en deux groupes disjoints. Le premier groupe est en rapport à *l'usage de sites web* comme : la date et le temps d'accès au site web, les pages web visitées et leurs acheminements et la taille d'un fichier téléchargé. Le deuxième groupe vise les informations concernant *les utilisateurs de sites web* comme les adresses IP, les fureteurs utilisés (Internet Explorer, NetScape, Mozila), le système d'exploitation (Linux, OS2, Windows), les cookies, les noms d'usagers si les pages de sites web qu'il a visité sont protégées par l'authentification htaccess et htpasswd [72, 26].

4.2.1.4 Le profil utilisateur

Il s'agit des informations constituant le profile des utilisateurs comme : l'âge, le sexe, la localisation géographique, la langue, la fonction, les objectifs et les pré-

férences. La collecte de certaines informations comme la localisation géographique et la langue pourrait être effectuée implicitement grâce aux fichiers journaux et aux cookies. Contrairement à l'âge, la fonction et les préférences, leur collecte devra se faire explicitement à travers un questionnaire à remplir en collaboration avec l'utilisateur [72, 26]. Il est important de dire que cette catégorie de données joue un rôle important dans la *modélisation des internautes*[9] [23].

4.2.2 Taxonomie de forage de web

Principalement, le forage du web est divisé en trois grandes catégories [75, 49, 30, 24] : *le forage du contenu*[10] c'est le processus d'extraction de connaissances à partir du contenu d'un document ou de sa description ; *le forage de la structure*[11] qui s'occupe de la structure organisationnelle du site web ; et enfin, *le forage de l'usage du site web*[12] qui a pour objectif d'extraire des connaissances cachées dans les fichiers journaux des serveurs web.

La figure 4.2 illustre la taxonomie du forage du web.

[9] Users modeling
[10] Web Content Mining
[11] Web Structure Mining
[12] Web Usage Mining

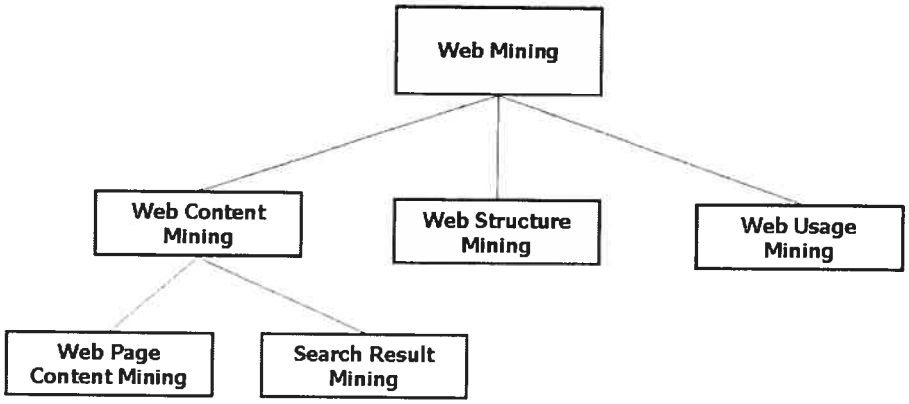

FIG. 4.2 – Taxonomie du forage du web [30]

4.2.2.1 Le forage de la structure

Cette catégorie se base principalement sur l'analyse des données décrivant la topologie d'un site web en vue de la réorganiser [49, 30, 75], par exemple, l'interconnexion des hyperliens. Dans cette catégorie, c'est-à dire, le forage de la structure [13], nous trouvons les algorithmes de classement utilisés par les moteurs de recherche comme PageRank que nous avons présenté au chapitre 2 ou HITS[52], SALSA [51] et CLEVER[20].

4.2.2.2 Le forage du contenu

Cette catégorie consiste à extraire des modèles appelés parfois motifs en se basant sur le contenu des documents d'un site web [49, 30, 75]. Le forage du contenu peut être divisé en deux catégories (voir figure 4.2). La première catégorie est *le forage du contenu des pages*[14], elle consiste à intégrer et organiser les données hétérogènes et semi-structurées du web en données plus structurées comme les données relationnelles, et accessibles pour les outils de datamining et

[13] Web Structure Mining
[14] Web Page Content Mining

les langages d'interrogation de données. Plusieurs travaux ont été menés dans ce sens, nous citons WebOQL [8], WebML [80], W3QL [47]. La deuxième catégorie comporte le développement de systèmes sophistiqués de l'Intelligence Artificielle qui peuvent agir de façon autonome ou semi autonome pour découvrir et organiser les informations sur le Web. Dans cette catégorie, nous trouvons l'agent de recherche intelligent[15] FAQ-Finder [35], le système BO (Bookmark Organizer) [52] en filtrage d'information[16] et Syskill & Webert [61], un agent web personnalisé[17]

4.2.2.3 Le forage de l'usage

Cette catégorie s'intéresse à l'usage de sites web en fouillant les traces laissées par les internautes dans les fichiers journaux [48, 29, 74]. Plusieurs travaux d'ordre général ont été réalisés comme WebSift [74] et SpeedTracer [78]. Le forage de l'usage (Web Usage Mining) peut servir en outre plusieurs domaines, tel qu'illustré à la figure 4.3.

[15]Intelligent Search Agents
[16]Information Filtering/ Categorization
[17]Personalized Web Agent

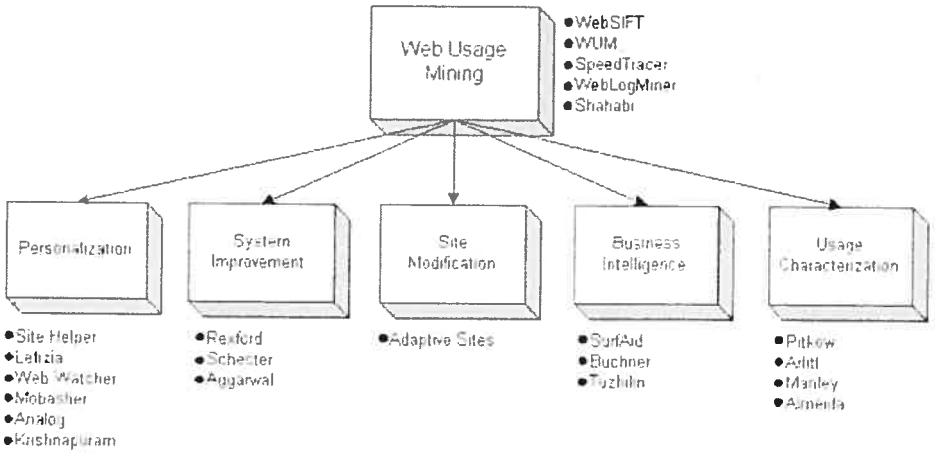

FIG. 4.3 – Domaines d'application du Web Usage Mining [72]

1. Amélioration des systèmes

L'un des objectifs de forage de l'usage est de développer des politiques permettant de comprendre comment les utilisateurs se servent de sites web pour identifier par exemple : des failles de sécurité, des intrusions, des fraudes, etc. [72].

Plusieurs travaux ont été réalisés dans ce sens, nous citons : Schechter [67] et Almeida [5].

2. Modification de sites web

La satisfaction des besoins des utilisateurs représente un élément important pour les sites Web. En effet, le forage de l'usage permet la restructuration du site web afin de le rendre attractif en s'appuyant sur les traces laissées par les utilisateurs [72].

3. Intelligence d'Affaires

Le forage de l'usage permet de révéler certaines questions importantes, à l'égard de l'usage des sites web, notamment, les sites commerciaux.À titre d'exemple : Qu'est ce qui a attiré l'utilisateur sur le site web ? Quel est le profil de l'utilisateur rentable ? Comment garder les utilisateurs pour toujours ? [72].

Parmi les travaux qui ont été faits dans ce domaine, nous citons : SurfAid Analytics [74] et Buchner [17].

4. Caractérisation d'usage

Le forage de l'usage pourrait être utilisé pour révéler des statistiques concernant les clics des utilisateurs. Par exemple, des modifications ont été apportées au fureteur Xmozaic, à l'Institut de Technologie de Géorgie afin d'enregistrer toutes les activités qui ont lieu sur la machine de l'internaute, c'est-à-dire, un fichier journal coté client retraçant les activités (Back/Forward, Downloading files, adding to bookmarks, etc.)[72].

Plusieurs travaux ont été réalisés dans ce sens : Pitkow [42] et Manley [54]

5. Personnalisation

La personnalisation [49, 59, 26, 75] a été définie comme étant l'utilisation d'analyses prédictives sur les données clients, pour conduire à une livraison ciblée d'informations ou des messages promotionnels. Elle consiste aussi à l'emploi de la technologie afin d'intégrer des informations personnelles à la présentation et au contenu de tout contact avec client ou partenaire, en taillant le contenu, la publicité et les services sur mesure pour des individus ou des groupes d'individus spécifiques.

En résumé, la personnalisation vise à tailler sur mesure les exigences d'un individu ou d'un groupe d'individus selon les informations recueillies sur ces personnes.

La personnalisation est devenue primordiale pour les concepteurs des sites web en général et pour les moteurs de recherche en particulier. Le travail que nous avons proposé se situe dans cette catégorie, c'est-à-dire, en forage de l'usage (WUM) et spécifiquement la personnalisation. Parmi les travaux ayant été réalisés dans cette direction :

WebLogMiner : Zaine *et al.* [82] ont proposé ce système, qui utilise l'outil d'analyse OLAP multidimensionnel pour analyser et explorer les données des fichiers

journaux, en vue de comprendre le comportement des visiteurs. Chaque dimension de cette analyse est caractérisée par plusieurs attributs, à titre d'exemple : l'URL demandé, type de fichier, type de fureteur, origine de requête représentent les attributs de la dimension : *fichier journal.*

SurfLen : Ce système a été proposé par Fu *et al.* [29]. SurfLen utilise le regroupement par affinité, c'est-à-dire, la technique des règles d'association pour la recommandation des pages web potentiellement intéressantes aux utilisateurs.

PageGather : Ce système a été proposé par Perkowitz et Etzioni [63], il utilise la matrice de co-occurrence de toutes les pages visitées d'un site afin de trouver des groupes de pages web qui sont fréquemment visitées durant une même session.

WebCANVAS : Cadez *et al.* [18] ont proposé un système utilisant le modèle de Markov pour visualiser les groupes des usagers similaires en vue de la recommendation.

Webace : Han *et al.* [37] ont proposé un système de catégorisation des documents sur le web. Après avoir catégorisé un ensemble de documents, Webace génère de nouvelles requêtes afin de mettre la main sur de nouveaux documents, en utilisant pour cela des techniques de regroupement[18]. Les documents retrouvés seront filtrés en ne gardant que ceux qui sont fortement liés avec l'ensemble des documents de départ.

WebWatcher : Ce système a été conçu par Armstrong *et al.* [7]. Il se base sur l'évaluation des utilisateurs. WebWatcher interprète la requête de l'usager dès sa réception en lui suggérant ensuite des liens et en tenant compte des évaluations que cet utilisateur a fait sur d'autres liens. Il garde une base de donnée dans laquelle sont emmagasinées des requêtes, des URLs et des évaluations faites par des utilisateurs, il utilise la technique TFIFD [19] pour mesurer la similarité entre les préférences des utilisateurs et les hyperliens de la page web en cours.

[18] clustering

[19] Term Frequency Inverse Document Frequency

SiteHelper : Ngu et Wu [57] ont proposé un système qui apprend les préférences de l'usager en regardant les pages web visitées par celui-ci. Le système extrait ensuite une liste des mots clés à partir des pages web sur lesquelles l'utilisateur a passé beaucoup de temps. Cette liste sera utilisée pour les recommandations futures.

4.2.3 Processus du forage de l'usage

La figure 4.4 montre les étapes du processus de forage du web [23, 24], qui doivent être réalisées dans d'ordre de leur présentation : 1) *le prétraitement* [20] consiste à collecter les données web, puis identifier toutes les transactions, les intégrer, les préparer et les faire passer à la seconde étape ; 2) *la découverte des modèles*[21] repose sur le choix des techniques de forage de données en vue d'extraire des modèles et des connaissances ; 3) *l'analyse et Evaluation des modèles*[22] a pour but de valider des résultats emportés dans l'étape précédente en utilisant pour cela des outils de visualisation comme : OLAP, SQL-Link et WebTools.

[20] Preprocessing
[21] Mining Algorithms
[22] Patterns analysis

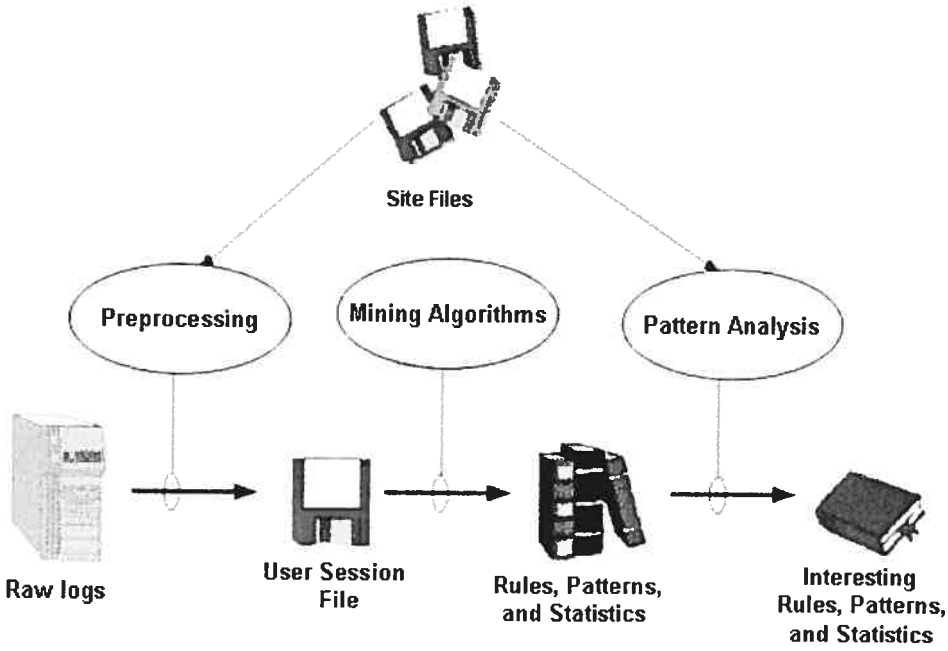

FIG. 4.4 – Processus du forage de l'usage [23, 75, 72]

4.2.3.1 Prétraitement

Cette étape consiste à réunir toutes les données nécessaires pour atteindre l'objectif imposé par une entreprise à priori. De toute évidence, ces données proviennent habituellement des fichiers journaux recueillant l'usage de sites web à l'insu des utilisateurs comme : l'adresse IP de l'ordinateur de l'utilisateur, la date et l'heure d'accès, le nom de l'utilisateur, les pages web visitées. D'autres informations relatives aux utilisateurs peuvent y être considérées, par exemple : le sexe, l'âge, la fonction, la ville, le pays, etc.

Après avoir ramassé ces données, il est important de nettoyer celles jugées superflues. Il s'agit précisément de requêtes concernant les pages web graphiques, des scripts dynamiques (php, cgi, pl, asp, etc.) et les requêtes laissées par les robots sur les sites web lorsqu'ils y passent pour les indexer. Il faudrait également iden-

tifier les transactions en provenance des serveurs proxy et reconnaître l'identité des utilisateurs cachés derrière ceux-ci, en faisant recours aux concepts de cookies et de sessions [23].

4.2.3.2 Recherche des modèles

L'objectif de cette étape consiste à appliquer les algorithmes de forage de données sur des données collectées en vue de les transformer en connaissances. Autrement dit, de donner une signification et une interprétation à ces données. Les techniques les plus connues sont [75] : a) les règles d'association ; b) la segmentation ; c) la classification ; d) les règles des séquences ; e) l'analyse de chemins.

a. Les règles d'association

La technique des règles d'association consiste à identifier les dépendances et corrélations existantes entre les caractéristiques observées sur un ensemble de données [72, 59, 27, 75]. En forage du web, cette analyse vise à déterminer les dépendances entre les pages de site web en vue de le reconstituer ou d'expliquer des événements.

Exemples

- **75%** des internautes localisés sur /Cne/prods/speciaux.htm ont fait des achats sur la page /Cne/prods/produit1.htm ;
- **50%** des internautes qui ont visité /Cne/prods/produit1.html ont également visité /Cne/prods/produit2.html et /Cne/prods/produit3.htm.

b. Le groupement

Il s'agit de créer des groupes homogènes à partir des données hétérogènes, inconnues au préalable. Ces groupes[23] seront interprétés afin de leurs donner signification et intérêt. En forage du web, l'utilité de cette analyse est de rallier

[23]clusters

les utilisateurs ayant des comportements similaires en espérant créer des clusters homogènes qui seront utiles pour de futures prédictions [72, 75].

Exemples

- **65%** des internautes qui ont fait un achat sur /Cne/prods/produit1.html sont des montréalais ;
- **50%** des internautes ayant acheté le *produit*1 sont des gens de 30 à 50 ans, dont **67%** sont des hommes.

c. Le classement

Il s'agit d'examiner les caractéristiques d'un utilisateur nouvellement représenté afin de l'affecter à une classe déjà prédéfinie. Ces classes peuvent être établies par les techniques de forage de données présentées à la section 3.7 comme par exemple, les arbres de décision, le raisonnement à base de cas et éventuellement l'analyse des liens présentée ci-dessous [72, 75].

d. L'analyse des liens[24]

Ce type d'analyse à pour but de déterminer et d'analyser les itinéraires empruntés par les utilisateurs sur un site web en vue de réorganiser la hiérarchie. Cette analyse se base fermement sur la *théorie de graphe*[25] [72, 75, 14].

Exemples

L'analyse des fichiers journaux a montré que :

- **65%** des internautes ont quitté le site après avoir visité au plus 7 pages web ;
- **60%** des visites sont localisées sur /Cne/prods/produit1.html.

De nombreuses techniques peuvent être mises en œuvre pour servir cette analyse. En voici quelques-unes : les statistiques traditionnelles, les réseaux Bayésiens et les règles d'association.

[24] Links analysis
[25] Graph Theory

e. Les règles des séquences[26]

Cette analyse permet d'examiner les activités des utilisateurs en se basant sur les transactions enregistrées sur les fichiers journaux, et en tenant compte du facteur temps, ce qui veut dire, que cette analyse se limite à un laps de temps bien déterminé [t1, t2]. Certains sites marchands profitent de cette analyse pour anticiper et prédire les besoins de ses clients. Cette analyse pourrait aussi être efficace pour optimiser le référencement en analysant évidemment les mots clés utilisés par les utilisateurs pendant leurs recherches [72, 75].

Exemple

- **50%** des clients ayant visité le lien /Cne/prods/index/produits.html ont fait des recherches la première semaine du mois de décembre 2004, sur Google.com par les mots clés $m1, m2, ...m3$.

4.2.3.3 Analyse des modèles

Cette phase nécessite la possession de certains outils de visualisation et de statistiques comme : Excel, SQL, SQL-Link, WebTool et OLAP. Il s'agit d'interpréter et d'évaluer les modèles et connaissances obtenus précédemment, en vue d'exploiter ceux qui sont pertinents à la prise de décision, et d'éliminer ceux qui ne sont pas rentables. Il est également possible de raffiner ces modèles, en modifiant les méthodologies établies antérieurement. Par exemple, il est possible de supprimer les groupes marginaux dans la tâche de groupement. Il est parfois envisageable d'incorporer d'autres sources de données ou de supprimer certains paramètres afin d'aboutir à des résultats satisfaisants.

[26] Sequence analysis

4.3 Conclusion

Dans ce chapitre, nous avons évoqué certains aspects visant surtout à clarifier le forage du web et ses catégories : WCM (Web Content Mining), WSM (Web Structure Mining) et WUM (Web Usage Mining), tout en évoquant les différents types de données du web : les données du contenu, les données de la structure, les données de l'usage et les données du profil d'utilisateur. Rappelons que notre travail consiste à réaliser un moteur de recherche personnalisé, utilisant conjointement les règles d'association abordées dans la section 3.7.6, et les requêtes traditionnelles formulées en langage d'interrogation de données SQL pour réaliser la recommandation, c'est-à-dire, pour fournir aux utilisateurs des résultats plus raffinés en se basant sur les similarités possibles entre eux.
Par conséquent, nous nous retrouvons dans un environnement très complexe, où il va falloir trouver un moyen pour garder quelque part, les traces de tout utilisateur utilisant BLED. Cette mission aurait été difficile (voire impossible) à atteindre si nous avions utilisé les cookies ou les fichiers journaux :

- Il est difficile de dévoiler l'identité des utilisateurs cachés derrière les serveurs proxy, ou qui se connectent à travers des adresses IP dynamiques ;
- Le refus ou la suppression des cookies complique la tâche des serveurs web. En effet, l'utilisateur qui efface ou refuse la reception les cookies sera toujours considéré comme un nouvel utilisateur à chaque fois qu'il se connecte sur le site web.

Afin d'éviter le problème d'identification des internautes utilisant BLED. Nous avons préféré l'utilisation des bases de données pour intercepter automatiquement les transactions faites, où chaque utilisateur doit introduire ses coordonnées (son pseudonyme et son mot de passe) pour qu'il soit reconnu par le système. Cette opération donnera, évidemment, plus d'exactitude et de précision vis-à-vis des

informations recueillies.

Le prochain chapitre détaillera notre solution intitulée BLED qui utilise les évaluations des documents faites par des utilisateurs afin de leur recommander les meilleurs documents. Seul l'utilisateur ayant lu un document peut juger de sa valeur.

Chapitre 5

Architecture de BLED

Ce chapitre représente l'élément clé de ce mémoire, nous y décrivons les concepts et les idées permettant de justifier l'originalité de notre approche, nous détaillons ensuite les principaux composants de notre système que nous avons appelé BLED, un système qui recommande les meilleures ressources web appréciées par les utilisateurs à travers l'Internet

5.1 Exigences

Rappelons que l'objectif principal est de concevoir un système qui répond aux exigences suivantes :

- BLED doit permettre aux utilisateurs d'économiser le temps consacré à la recherche d'informations sur le web ;
- BLED doit fournir aux utilisateurs une stratégie leur permettant d'organiser et de stocker leurs documents favoris, dans leurs propres dossiers

pour des consultations ultérieures. Cette fonctionnalité est très importante car il arrive souvent à des utilisateurs d'oublier comment ils ont fait leurs recherches pour trouver des documents intéressants. En outre, BLED s'occupe automatiquement de la gestion des liens (URLs) de ses utilisateurs, en vue de ne garder que ceux qui sont opérationnels et élimine ceux qui sont périmés ;

- BLED doit permettre aux utilisateurs inexpérimentés d'apprendre comment faire une recherche ciblée. Par exemple, un historien qui veut acheter un micro-ordinateur, il introduira peut-être une requête simple comme *"Achat Pentium 4"*, alors qu'un spécialiste en matériel informatique, qui a certainement plus de connaissances en matériels informatiques, formulera une requête plus complexe, par exemple, *"Achat Pentium 4 3.6MHz Carte mère Asus 512 DDR, 60 Go 7200rpm 8Mo mémoire cache"* , cette requête est plus précise en terme de qualité par rapport à celle recherchée par l'historien. Par conséquent ce dernier peut apprendre du spécialiste ;
- BLED doit partager et recommander des ressources aux utilisateurs en profitant de leurs recherches pertinentes. Ceci constitue le coeur de notre approche, qui se base principalement sur la technique des règles d'association permettant de trouver les associations entre ressources précédemment appréciées entre utilisateurs et de leur recommander les plus intéressantes.

5.2 Scénario d'utilisation

Le scénario que nous allons présenter dans ce paragraphe a pour objectif de faire la lumière sur le fonctionnement général de notre système BLED. Ce scénario est illustré à la figure 5.1.

Tout utilisateur possédant une connexion Internet et un fureteur[1] Web peut utiliser BLED. C'est un moteur de recherche semblable aux autres moteurs de recherche traditionnels comme Google, Altavista et Teoma. Cependant, il fonctionne selon deux modes différents.

5.2.1 Le mode anonymat

L'utilisateur n'a pas besoin de s'identifer pour utiliser BLED, il peut faire ses recherches librement, il peut également bénéficier de certaines fonctionnalités offertes par BLED, mais il ne pourra pas avoir son propre dossier personnel qui lui permet d'organiser ses documents favoris.

5.2.2 Le mode personnalisé

Dans ce mode, l'utilisateur doit s'identifier pour pouvoir utiliser BLED. L'avantage de ce mode c'est que l'utilisateur peut profiter de toutes les fonctionnalités offertes par BLED.

[1] Browser

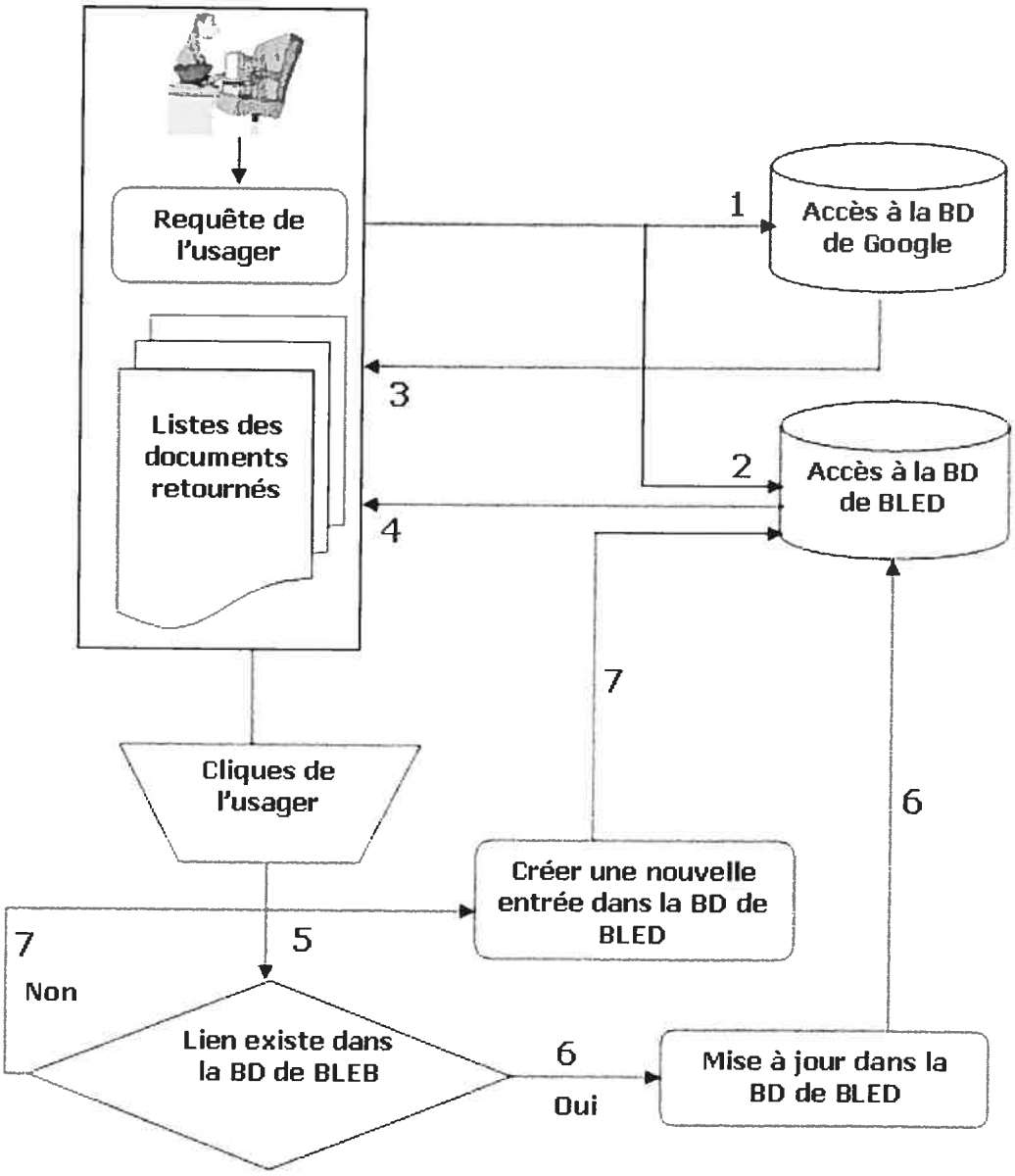

FIG. 5.1 – Scénario d'utilisation

Supposons qu'un utilisateur nommé "MYBLED" veut utiliser BLED en **mode personnalisé**, il est absolument nécessaire qu'il soit inscrit dans la base de données des utilisateurs pour pouvoir naviguer en utilisant ce mode.

5.2.2.1 Phase d'authentification

Après avoir introduit son Pseudonyme (Exemple : "MYBLED") et son mot de passe. Le système BLED attribue immédiatement une session à cet utilisateur, qui sera active jusqu'à ce que l'utilisateur "MYBLED" se déconnecte ou ferme son fureteur. L'objectif d'une session est de garder ses recherches (mots clés et liens) pour une utilisation ultérieure. L'utilisateur "MYBLED" peut également gérer son dossier, c'est-à-dire, d'évaluer et d'effacer des liens de son dossier personnel, ou consulter et trier des liens selon des critères qu'il aurait souhaités.

5.2.2.2 Phase de recherche

Admettons que l'utilisateur "MYBLED" fasse une simple recherche sur les mots clés "web mining". Celui-ci peut approfondir sa recherche en précisant le type de document (PDF, PPT, DOC, etc.), il peut aussi la consolider au moyen des opérateurs : (+ : réclame la présence des mots clés dans les documents trouvés), (- : exclut les documents contenant les mots clés), (~ : inclut les documents contenant les synonymes des mots clés). L'utilisateur "MYBLED" choisit le format PDF pour cette requête (voir figure 5.2).

Une fois que l'utilisateur "MYBLED" clique sur le bouton "OK", BLED lance deux programmes indépendants et lui envoie les résultats suivants :

a. Résultats de la base de données de Google

Ce programme interroge la base de données de Google via son API qui est basée sur le protocole SOAP et le langage WSDL (voir figure 5.1 (1)). Le programme reçoit en entrée des paramètres comme par exemple : "web mining" et "PDF". L'API Google fournira ainsi la liste des documents (voir figure 5.1 (3)) s'il y a lieu, dont seules les références contenant les mots clés de l'expression "web mining" seront affichés (voir figure 5.2 : Accès à la BD de Google). Ces références sont classées selon une politique de classement propre au moteur de recherche Google.

b. Résultats de la base de données de BLED (Approche 1)

Ce programme recherche localement la requête "web mining", c'est-à-dire, dans la base de données de BLED (voir figure 5.1 (2)) en utilisant la technique de corrélation entre mots clés recherchés. C'est grâce au langage d'interrogation de données MySQL que cette tâche est réalisée. Par exemple, la figure 5.2 à gauche de l'écran indique qu'il y exactement 10 liens en réponse à la requête "web mining" demandée par l'utilisateur "MYBLED". Ces liens sont sélectionnés parmi 46 liens visités par les utilisateurs de BLED d'après le code MySQL suivant :

```
mysql> SELECT count(*) FROM data WHERE user_request LIKE '\%web\%'
or '\%mining\%';
 +----------+ | count(*) | +----------+
 |       46 |
+----------+ 1 row in set (0.03 sec)
```

Rappelons que pour qu'un document soit sélectionné il doit être évalué à une moyenne supérieure ou égale à 5/10 par l'ensemble des utilisateurs. Une moyenne que nous avons jugée acceptable pour qu'un document soit fiable (recommendable).

La figure 5.2 montre que BLED recommande à l'utilisateur "MYBLED" 10 liens ayant été appréciés par d'autres utilisateurs. Ceci, lui permet aussi d'apprendre comment inférer les mots clés convenables pour affiner sa recherche. Par exemple, il aperçoit à gauche de l'écran 4 nouvelles requêtes qui pourraient lui être utiles. Ces requêtes sont "Web Mining in Soft Computing", "these data mining", "Data mining for corporate masses" et "Parallel Association Rule Mining". Toutes ces requêtes comprennent au moins le mot clé "web", le mot clé "mining" ou les deux ensembles, c'est le principe de *similarité entre mots clés.*

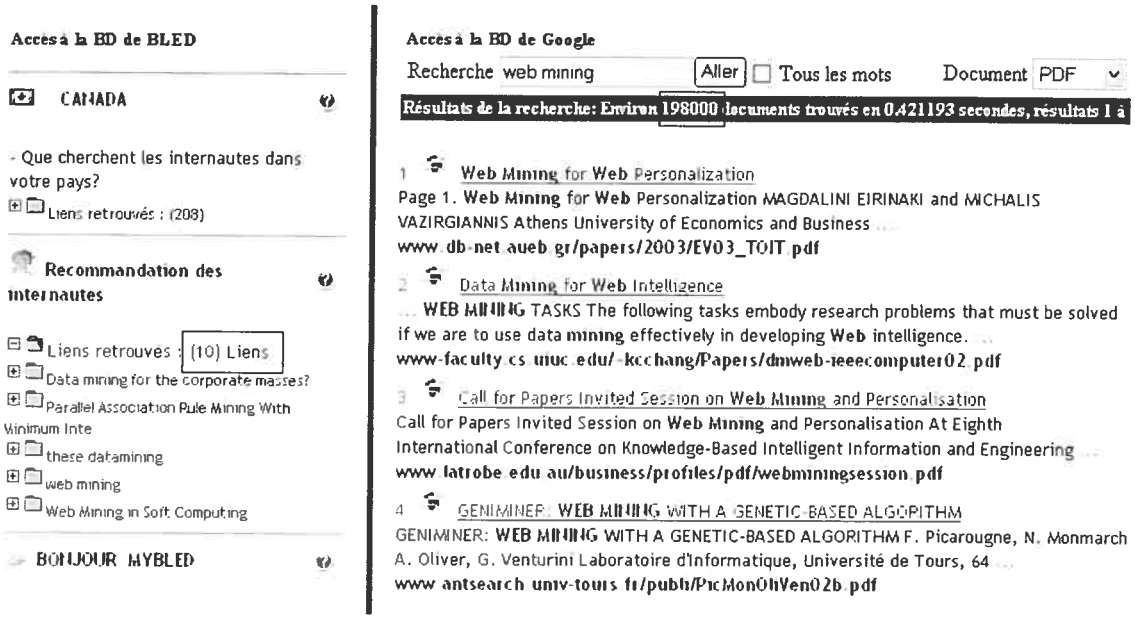

FIG. 5.2 – Recommandation de BLED : Approche 1

c. Résultats de la base de données de BLED (Approche 2)

Examinons l'exemple de la figure 5.3, l'utilisateur "MYBLED" recherche cette fois-ci seulement les documents contenant le mot clé "data" ayant l'extension "PDF". BLED lui suggère les documents renvoyés par la base de données de Google tel qu'il est montré à la 5.3. Un chiffre d'environ 3290000 documents à explorer, il lui propose aussi deux autres listes, dont la première comprend 7 liens (voir 5.3 : Accès à la base de données de BLED) présentant la recommandation des internautes basée sur la similarité de mot clé "data", et l'autre liste contient 4 liens suggérant la recommandation fondée sur une technique très utilisée en data mining appelée "technique des règles d'association" ou parfois "groupement par affinité" ou "l'analyse de panier de la ménagère". Cette technique a été détaillée dans la section 3.7.6.

La figure 5.3 montre que BLED a trouvé dans sa base de données des règles d'association, une règle affirmant que l'utilisateur "MYBLED" a certainement cliqué sur un ou plusieurs liens, qui forment les (conditions) premisses d'une règle d'association qui donne les 4 documents comme conséquences (résultats).
Ce qui est remarquable dans cet exemple c'est que les 4 liens ont un rapport direct avec le mot clé "data" recherché par l'utilisateur "MYBLED". Par exemple, le document"http ://www.stat.ucl.ac.be/ISrapport/rap00/rapportfr2000.ps" correspond à une recherche de la requête "datamining" faite par un autre utilisateur.
Parmi les avantages du mode **personnalisé** c'est qu'il est possible à l'utilisateur "MYBLED" de partager des liens avec un autre utilisateur, il suffit pour cela de saisir son adresse de courrier électronique (voir 5.3).

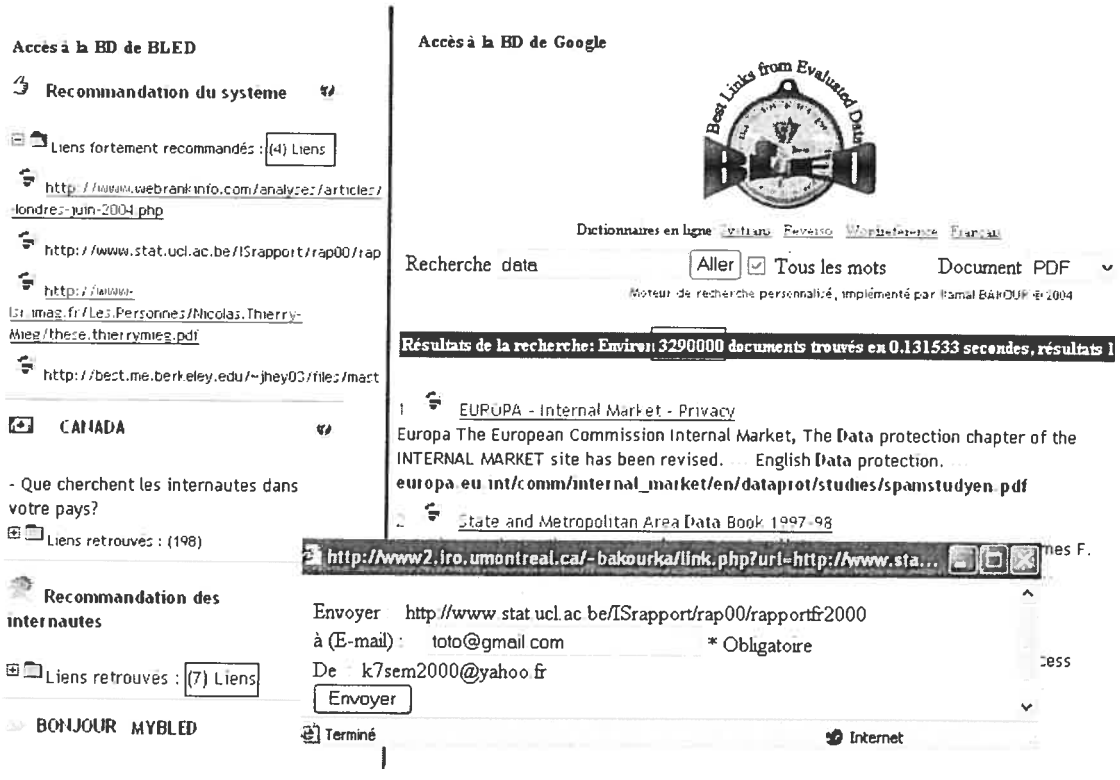

FIG. 5.3 – Recommandation de BLED : Approche 2

5.2.2.3 Traces de l'usager "MYBLED"

En ce qui concerne le stockage et la mise à jour des recherches de l'utilisateur "MYBLED". Deux cas peuvent se présenter lorsque celui-ci clique sur un lien :

- Le lien a déjà été visité par l'utilisateur "MYBLED" durant des sessions antérieures. BLED met alors à jour dans son propre dossier le nombre de visites de ce document et la date de sa dernière visite (voir figure 5.1(6)) ;
- Le lien n'a jamais été visité par l'utilisateur "MYBLED". BLED crée une nouvelle entrée dans son dossier (voir figure 5.1(7)).

Le tableau 5.1 montre certains attributs d'un enregistrement dans le dossier de l'utilisateur "MYBLED".

TAB. 5.1 – Exemple du simple enregistrement

N	Attribut	Valeur
1	requête	web mining
2	Lien	http ://maya.cs.depaul.edu/ mobasher/papers/webkdd2000.pdf
3	Vote	5/10
4	Durée de visite	45.98 seconds
4	Nombre de visites	3
5	format de liens	PDF
6	langue	Français
7	Adresse IP	69.70.149.152
⋮	⋮	⋮

5.3 Architecture de BLED

Le système BLED est constitué de deux principaux modules : *le module en ligne* s'exécute en temps réel en interaction avec les utilisateurs. Il a pour but de mettre en place une session entre l'utilisateur et BLED afin d'identifier toutes ses empreintes pendant cette session. Nous avons choisi cette solution pour éviter les problèmes liés aux cookies et aux fichiers journaux des serveurs proxy tels que décrits dans la section 4.1.1 et la section 4.1.2. En outre, ce module est destiné au traitement de requêtes des utilisateurs exprimées par des mots clés ; *le module hors ligne* représente l'élément clé de notre vision, il sert à l'extraction des règles d'association en se basant sur la technique des règles d'association.

L'architecture du système BLED est illustrée à la figure 5.4 :

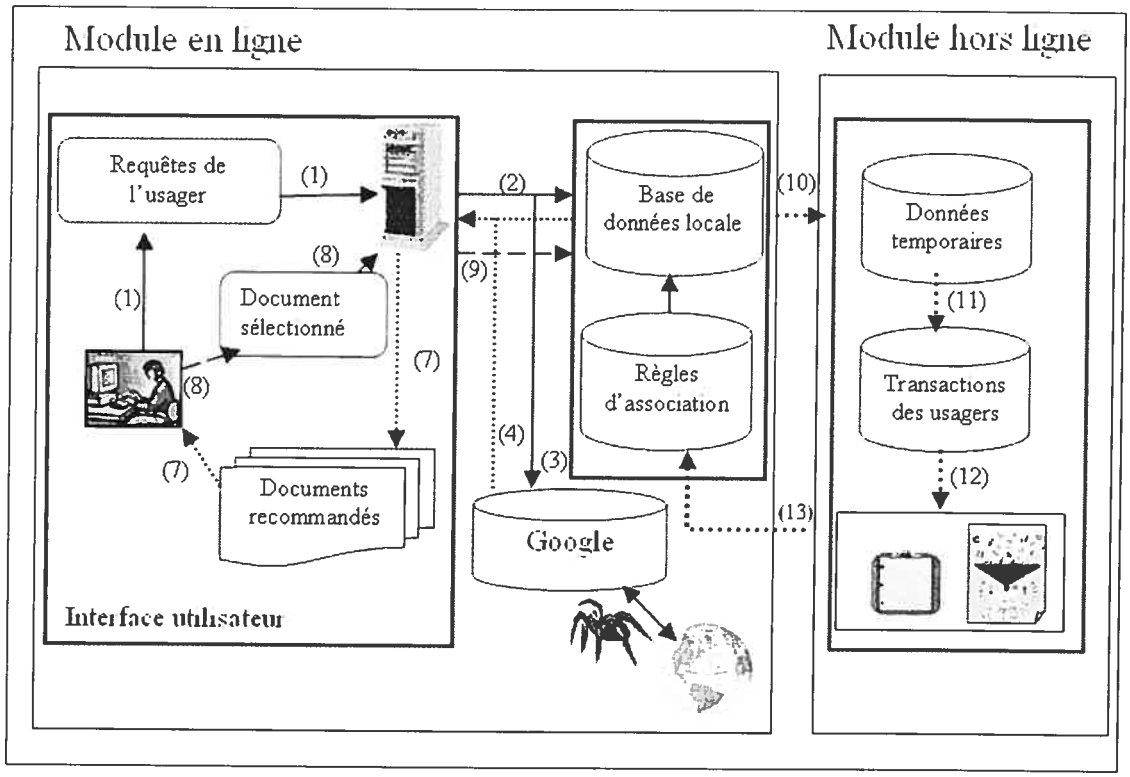

FIG. 5.4 – Architecture générale de BLED

5.4 Module hors ligne

Comme nous l'avons déjà présenté, la technique des règles d'association a pour but d'établir les associations (affinités) existantes entre documents, antérieurement appréciés par les usagers du système BLED, afin de pouvoir plus tard trouver lesquels seraient susceptibles d'aider les utilisateurs au cours de leurs recherches. Ce module doit souvent s'exécuter hors ligne, car il a un temps d'exécution très coûteux lorsqu'il utilise l'algorithme APRIORI. Cette opération se fait selon les étapes de la figure 5.5.

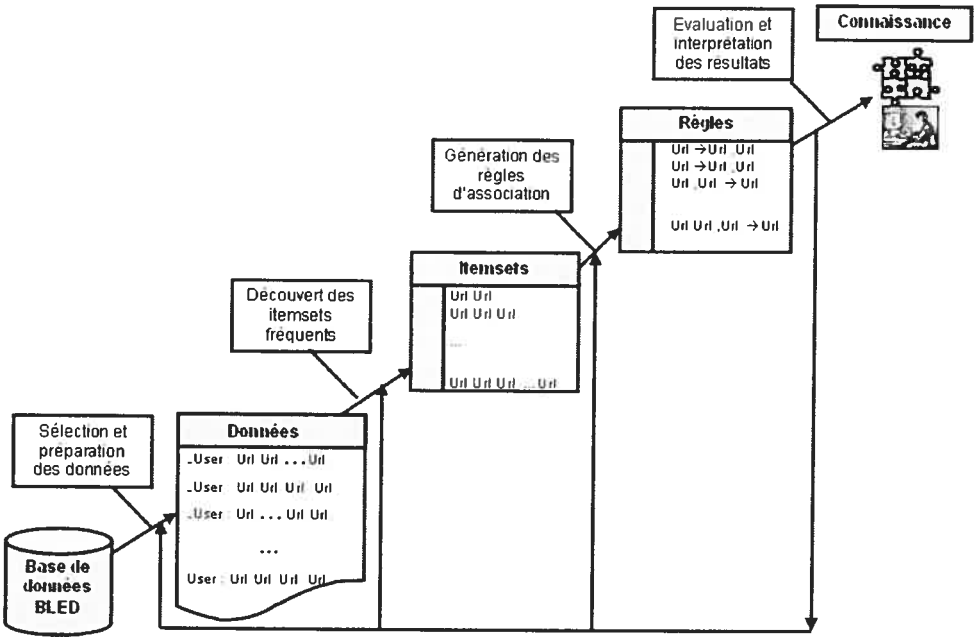

FIG. 5.5 – Cycle de génération des règles d'association

La procédure générale du module hors ligne est comme suit :

Algorithm 5 Programme de génération des règles d'association

Entrée: $\mathcal{B}$: Base de données BLED ; `minsupport` ; `minconfiance` ;
Sortie: Nouvelle table des règles d'association ; Copie de protection de la base de données de BLED ;

Début
`Check()` ; // Inspections des liens de la base de données de BLED
`Backup()` ; // Sauvegarde de la base de données de BLED
`TransID()` ;//Indentification des transactions des utilisateurs.
Apriori(minsupport,minconfiance) ;// Génération des règles d'association.
`Rules()` ; // M.A.J des règles d'association dans BLED.
Retourner Copie de secours de données de BLED ;
Retourner Nouvelle table des règles d'association ;
Fin

1. `Check()` : BLED met en place une fonction qui permet d'inspecter régulièrement l'état de chaque lien (URL) stocké dans sa base de données. Lors de cette vérification, la fonction `Check()` retourne deux états possibles pour chaque lien :
- Lien est toujours actif (présentement opérationnel) ;
- Lien a déjà été marqué temporairement inactif (lien mort) depuis la dernière inspection. Ce lien redevient opérationnel s'il est rétablit et il sera supprimé physiquement de la base de données dans le cas contraire. Notons que le système BLED ne tient pas en compte des liens morts au court des traitements.

2. `Backup()` : cette fonction permet de créer une copie de sécurité de la base de données BLED.

3. `TransID()` : permet de générer la matrice des transactions de tous les utilisateurs de BLED inscrits dans la base de donnée, c'est-à-dire, d'identifier pour chaque utilisateur quels sont les documents qu'il a appréciés (voir figure 5.5 :

Données).

4. `Apriori(`*minsupport*`,`*minconfiance*`)` : permet de générer toutes les règles d'association en utilisant la matrice des transactions obtenue par la procédure `TransID()`. Les valeurs des variables *minsupport* et *minconfiance* sont respectivement 7% et 50%. Le choix de ces valeurs est subjectif, dans notre cas par exemple, nous avons choisi un *minsupport* de 7% pour garder le maximum de documents (voir figure 5.5 : **Itemsets, Règles**).

5. `Rules()` : cette fonction a pour objectif de mettre à jour la table des règles d'association dans la base de données de BLED.

5.4.1 Sélection et préparation des données

Cette étape représente la première phase du processus d'extraction des règles d'association. Elle consiste à intégrer, nettoyer et transformer les données des utilisateurs de BLED.

5.4.1.1 Table des URLs les plus appréciées

Cette opération (figure 5.4 action 10) permet d'éliminer tout document (URLs) ayant reçu une évaluation moyenne inférieure à 5/10 en ne gardant que ceux dont la moyenne est supérieure ou égale à 5/10. Le tableau suivant montre un extrait de la table des documents les plus appréciés. Chaque URL est codifié par un identifiant unique.

TAB. 5.2 – Extrait de la table des liens pertinents

Code URL	Lien (URL)	Evaluation moyenne des liens
001	http ://www.umontreal.ca/	7.33/10
002	http ://myjeeves.ask.com/	5.25/10
003	http ://www.fooxx.com/	6.33/10
004	http ://myjeeves.ask.com/	9.00/10
005	http ://miaif.lip6.fr/	5.00/10
006	http ://www.cookiecenter.com/	9.66/10

Exemple

– Le site de l'université de Montreal a été évalué à une moyenne de 7.33/10.

5.4.1.2 Matrice des transactions des utilisateurs

Une fois la table des URLs les plus appréciées créée. Une autre opération (figure 5.4 action 11) consiste à établir la matrice des transactions des utilisateurs, celle-ci constitue l'entrée fournie à l'algorithme APRIORI. Elle ne comporte dans ses cellules que deux valeurs, 0 ou 1, 1 si l'utilisateur a apprécié l'URL, c'est-à-dire, qu'il a évalué avec une note supérieure ou égale à 5, et 0 dans le cas contraire. Le tableau suivant fournit une illustration de cette matrice.

TAB. 5.3 – Extrait de la table d'identification des transactions usagers

$CodeUsager/URLs$	URL_1	URL_2	URL_3	URL_4	URL_5	...	URL_n
T000	1	0	0	1	0	.	0
T001	0	1	1	0	0	.	1
T002	1	0	1	0	0	.	1
T004	0	1	0	0	1	.	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T099	1	1	0	0	1	.	1

Exemple

- Dans l'exemple de la table 5.3, l'utilisateur T000 a apprécié les liens URL_1 et URL_4. Par contre, il n'a pas apprécié les liens URL_2, URL_3, URL_5 et URL_n

5.4.2 Découverte des items fréquents

Après avoir établi la matrice des transactions des utilisateurs, le module hors ligne exécutera **(l'algorithme 1)** (voir section 3.7.7) afin de découvrir tous les ensembles qui ont un support supérieur au *minsupport* (Dans notre cas nous avons pris 7%). Rappelons que le support d'un ensemble d'items correspond au nombre de transactions que contient cet ensemble d'items. Ces ensembles représentent les $k-itemsets$ fréquents qui seront ensuite exploités par **(l'algorithme 3)** (voir section 3.7.9) pour générer les règles d'association.

Exemple

Supposons que la table des URLs pertinents contient seulement 4 liens. La découverte des items fréquents consiste à générer une structure de données sous forme d'un treillis tel qu'il est illustré à la figure 5.6, les 4 liens engendrent alors $(2^4 - 1)$ candidats soit 15.

(0-itemsets) : $\{\}$

(1-itemsets) : $\{URL_1, URL_2, URL_3, URL_4\}$;

(2-itemsets) : $\{URL_1URL_2, URL_1URL_3, URL_1URL_4, URL_2URL_3, URL_2$
$URL_4, URL_3URL_4\}$;

(3-itemsets) : $\{URL_1URL_2URL_3, URL_1URL_2URL_4, URL_2URL_3URL_4, URL_1URL_3URL_4\}$;

(4-itemsets) : $\{URL_1URL_2URL_3URL_4\}$;

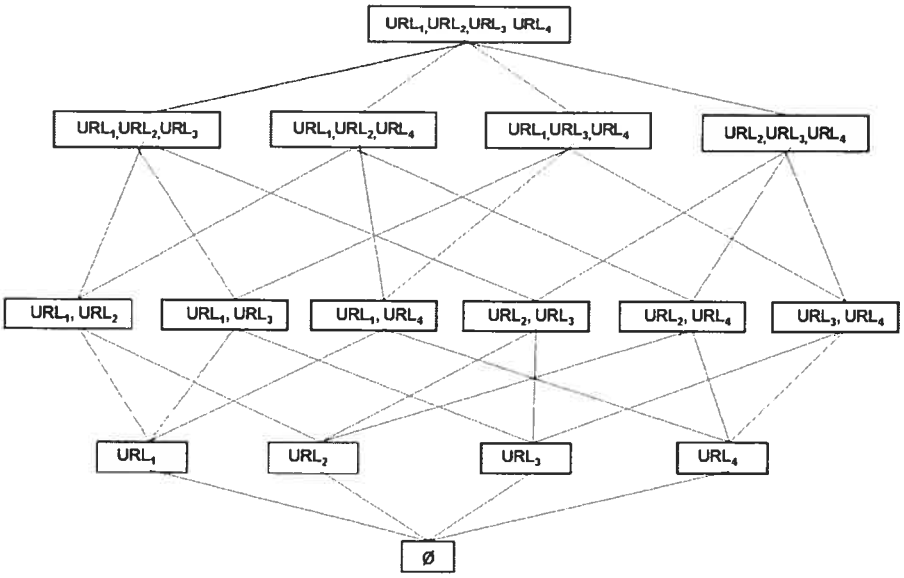

FIG. 5.6 – Exemple des itemsets

5.4.3 Génération des règles d'association

(**L'algorithme 3**) que nous avons montré dans la section 3.7.9 permet de générer les règles d'association, en utilisant tous les itemsets fréquents obtenus à l'étape précédente. Les règles sont générées à partir des itemsets fréquents et leurs supports associés. Pour générer les règles nous devons examiner pour chacun des $(k-itemsets)$ fréquents tous les $(k-1)itemsets$, puis ses $(k-2)itemsets$ jusqu'à $(1-itemset)$ en calculant la confiance de la règle. Par exemple, pour générer les règles d'association de l'itemset fréquent $F = \{URL_1URL_2URL_3URL_4\}$ nous devons examiner tous les sous-ensembles non vides de F, soient P. Pour chacun de ces sous-ensembles P, nous renvoyons une règle d'association de la forme : $\{R : P \Longrightarrow (F-P)\}$ si la valeur du rapport $\frac{support(F)}{support(P)}$ qui dénote la confiance de la règle vaut au moins $minconfiance$ (50% dans notre cas).

L'exemple suivant montre un prototype des règles d'association.

$$\{R_1 : URL_1 \Longrightarrow URL_2URL_3URL_4, (66\%)\}$$

L'interprétation de cette règle d'association est que 66% des utilisateurs qui ont visité URL_1 ont également visité URL_2, URL_3 et URL_4

5.4.4 Interprétation et évaluation des règles d'association

Il existe de nombreux outils qui ont été spécifiquement conçus pour l'évaluation et la visualisation des règles d'association, par exemple, *Rule Visualizer* qui permet la visualisation des règles d'association sous forme graphique ou textuelle en choisissant un échantillon de l'ensemble global des règles d'association. Lorsque les résultats obtenus par l'algorithme APRIORI ne sont pas satisfaisants, par exemple, des règles inexplicables comme la règle d'association du supermarché WALMART aux États Unis, qui disait "*Si achat bière alors achat couche-culotte*". Là, des enquêtes complémentaires ont été nécessaires afin de comprendre la signification de cette règle, bien que les taux de support et de confiance de cette règle étaient raisonnables. Dans certaines situations, il est difficile de prendre une décision et il devient absolument nécessaire de revoir le processus du forage de données. Par exemple, des modifications pourront être apportées sur les seuils minimaux du support $minsupport$ et de la confiance $minconfiance$, ou encore sur les critères de sélection des données concernant les utilisateurs et leurs documents pour améliorer la qualité des règles d'association générées.

5.4.5 Mise à jour de la base de données

Cette opération met à jour les règles d'association pertinentes dans la base de données de BLED. L'enrichissement de cette base se fait au fur et à mesure que la base de données principale de BLED progresse.

5.5 Module en ligne

Lorsque l'utilisateur saisit sa requête dans la case réservée à la recherche. BLED lance en parallèle deux processus action (2) et action (3) dans la figure 5.4, qui vont tenter de répondre à sa requête de deux façons différentes : par recherche locale et par accès à l'API de Google.

Algorithm 6 Le Module en Ligne

Entrée: `DB` : Base de données BLED, `Requete_Usager` : Ensemble de mots clés à chercher ;

```
Début
tant que Requete_Usager faire
  si Ouvrir_Session(Usager) alors
    BLED_Recherche(Requete_Usager, Usager);
    GOOGLE_Recherche(Requete_Usager);
    MAJ_BLED(Requete_Usager, Usager);
  sinon
    BLED_Recherche(Requete_Usager);
    GOOGLE_Recherche(Requete_Usager);
  fin si
fin tant que
Fin
```

5.5.1 Ouverture d'une session

Le système implémente un mécanisme d'identification **(algorithme 7)** qui permet à plusieurs utilisateurs de pouvoir l'utiliser simultanément, sans aucune interférence. Pour qu'un utilisateur puisse tirer avantage de toutes les fonctionnalités offertes par le système BLED, il doit s'identifier en indiquant tout d'abord son pseudonyme suivi de son mot de passe. Ces derniers lui auront été attribués au moment de la création de son compte. Pendant chaque session, le système

récoltera les traces des actions effectuées par les utilisateurs afin de les exploiter pour des fins de recommandation.

Algorithm 7 Ouvrir_Session

Entrée: DB : Base de données BLED; *Pseudonyme* : Nom de l'usager ;*MotPasse* : Mot de passe de l'usager ;
Sortie: Satut : booléen
Début
DB_Connect(DB) ;
si Usager_Existe(Pseudonyme, MotPasse) **alors**
Satut=vrai ;
sinon
Satut=Faux ;
fin si
Retourner *Statut* ;
Fin

5.5.2 Recherche locale

BLED interroge sa base de données locale (figure 5.4 action 2) comme suit :

Algorithm 8 BLED_Recherche

Entrée: Requete_Usager : Requête à rechercher ;
Sortie: BLED_Result_Pays, BLED_Result_Usagers, BLED_Result_RA : Structure contenant les URLs trouvées par fonction de BLED ;
Début
BLED_Result_Pays=BLED_Pays → Search(Requete_Usager) ;
BLED_Result_Usager=BLED_Usager → Search(Requete_Usager) ;
BLED_Result_RA=BLED_RA → Search(Requete_Usager) ;
Retourner BLED_Result_Pays ;BLED_Result_Usagers ;BLED_Result_RA ;
Fin

1. BLED_Usager → Search()**(algorithme 8)** examine le contenu de cette base afin de récupérer les documents correspondant aux requêtes similaires à celle fournie par cet utilisateur. Le critère de recherche consiste à trouver les documents

ayant déjà été évalués supérieurs ou égaux à 5/10, par d'autres utilisateurs et dont les requêtes sont similaires à celle fournie par l'utilisateur. Il est possible que ce processus ne retourne rien si sa recherche a été infructueuse.

Considérons le scénario où un utilisateur va chercher le mot "Java". Le processus de recherche locale va fouiller dans la base de données de BLED pour tenter de retrouver tous les documents cherchés et évalués supérieurs ou égaux à 5/10, par d'autres utilisateurs dont les requêtes sont proches de "Java". Le système pourra alors lui recommander les documents correspondants à des requêtes comme "Java Bean", "Javascript", "Java script", ce qui correspond de manière générale à des requêtes de type "%Java%". Pour l'instant, ce module n'utilise malheureusement pas des techniques issues d'extraction de l'information ou de text mining pour interroger la base de données de BLED, mais cela fait partie des améliorations considérées pour le futur. Pour le moment, il utilise uniquement SQL (Structured Query Language) comme méthode d'interrogation de la base de données de BLED. L'avantage principal de cette technique est qu'elle est facile à comprendre et simple à manipuler, mais elle a parfois comme inconvénient de retourner trop de résultats pour une simple requête. Aussi en se basant sur SQL le système ne peut pas se rendre compte à quel point deux requêtes sont proches ou différentes sémantiquement, par exemple dans le cas de requêtes comme "Data Mining", "Fouille de Données" ou "Forage de Données" le système ne pourra pas reconnaître la similarité de ces requêtes au niveau du sens.

2. `BLED_Pays` → `Search()` **(algorithme 8)** permet à l'usager de visualiser les liens (URLs) appréciés dans le pays où il effectue ses recherches. Par exemple, si l'usager se trouve au Canada, le système lui montre les meilleurs documents appréciés dans ce pays.

3. `BLED_RA → Search()` **(algorithme 8)** interroge la table des règles d'association pour recommander des documents aux usagers.

Exemple

Admettons que la table des règles d'association contient les enregistrements du tableau 5.4.

TAB. 5.4 – Table des règles d'association

N°	Premisse	Conséquence	Confiance
1	URL_1	URL_2URL_3	50%
2	URL_1URL_2	URL_3	75%
3	$URL_3URL_4URL_5$	URL_1URL_2	93%
4	URL_2URL_3	$URL_1URL_5URL_6$	55%
5	URL_6	$URL_2URL_4URL_5URL_8$	75%
⋮	⋮	⋮	⋮

Supposons qu'un usager clique sur *l'*URL_1. La fonction `BLED_RA → Search(`URL_1`)` examine le tableau 5.4 et lui recommande les documents URL_2 et URL_3. Dans cette règle, 50% des usagers ayant apprécié *l'*URL_1 ont également apprécié les URL_2 et URL_3.

Pour évaluer une règle d'association, il faut tenir en considération plusieurs indicateurs. Par exemple, examinons avec précision les règles d'association $\{R1 : URL_1 \Longrightarrow URL_2URL_3, 50\%\}$ et $\{R5 : URL_6 \Longrightarrow URL_2URL_4URL_5URL_8, 75\%\}$. Dès le premier regard, le lecteur constate que la règle $R5$ est plus puissante que la règle $R1$, car son niveau de confiance 75% est plus grand que 50% de la règle $R1$. Cependant, si nous analysons profondément la règle $R5$ nous constatons qu'il y a 4 utilisateurs ayant apprécié l'URL_6, dont 3 utilisateurs ont apprécié l'ensemble $URL_2URL_4URL_5URL_6$ et URL_8.

Par contre, pour la règle $R1$ il y a 14 utilisateurs ayant apprécié l'URL_1, dont

7 utilisateurs ont apprécié l'ensemble URL_1, URL_2 et URL_3. Par conséquent la règle $R1$ est mieux que la règle $R5$.

5.5.3 Accès à la base de données de Google

Le processus (figure 5.4 action 3)**(algorithme 9)** interroge la base de données du moteur de recherche Google. Ce processus permet de récupérer une structure contenant la liste des documents trouvés auprès de Google, en fonction de la requête demandée. Google fournit une API qui se base sur le protocole SOAP et le langage WSDL et qui permet d'effectuer gratuitement jusqu'à 1000 requêtes par jour. En effet, cette API permet de réaliser des applications qui utilisent la technologie basée sur l'algorithme PageRank et la base de données de Google. L'API Google reçoit comme entrée les requêtes provenant d'une application installée sur un site externe et fournit comme sortie une liste de documents sous forme d'objets structurés correspondant aux résultats d'une requête.

Algorithm 9 GOOGLE_Recherche

Entrée: `Requete_Usager` : Requête à rechercher ;
Sortie: `GOOGLE_Result` : Structure contenant les URLs retournées par SOAP Google ;

Début
`GOOGLE_Result=Google → Search(Requete_Usager)` ;
Retourner `GOOGLE_Result` ;
Fin

5.5.4 Mise à jour régulière de la base de données

Ce processus a pour mission d'enregistrer les clics des utilisateurs en suivant les étapes :

1. Rediriger le navigateur de l'utilisateur vers la ressource demandée (figure 5.4 action 9, **Algorithme 10**, `URL_Redirect()`).

2. Mettre à jour l'entrée correspondant à cet URL dans la base de données locale (figure 5.4 action 9, **Algorithme 10**, `MAJ_Enregistrement_DB()`) pour cet utilisateur si celle-ci a déjà été visitée antérieurement . La mise à jour portera sur certains attributs tels que le nombre de visites, la date de dernière visite et la durée de visite, etc.

3. Créer pour cet utilisateur une nouvelle entrée dans la base de données de BLED (figure 5.4 action 9, **Algorithme 10**, `Nouvel_Enregistrement_DB()`). Cette entrée va contenir l'identifiant de l'utilisateur et quelques informations se rapportant à l'URL lui-même. En pratique, cela se produira lorsque cet URL n'a jamais été vu auparavant par l'utilisateur actuel.

Algorithm 10 MAJ_BLED

```
Entrée: Ressource : Informations concernant l'usager la requête et l'URL ;
  DB : Base de données BLED ;
  Début
  URL_Redirect(URL) ; // Redirection de navigateur de l'usager à l'URL demandée
  si URL_Existe(DB, Usager, URL) alors
    MAJ_Enregistrement_DB(DB,Ressource) ;
  sinon
    Nouvel_Enregistrement_DB(DB,Ressource) ;
  fin si
  Fin
```

5.6 Conclusion

Dans ce chapitre, nous avons examiné en détail l'architecture et le fonctionnement de la solution BLED. Les systèmes de recommandations sont une solution efficace pour partager l'information entre utilisateurs. Certains systèmes utilisent le filtrage collaboratif [2] fondé sur les appréciations des utilisateurs partageant les mêmes goûts, d'autres systèmes utilisent le forage de l'usage [3] (voir section 4.2.2.3). Nous avons montré que les techniques mises en oeuvre dans BLED ont répondu à la problématique que nous avons présentée au début de ce travail. En effet, nous avons incorporé la technique des règles d'association pour recommander des documents susceptibles d'intéresser d'autres utilisateurs, d'autre part, nous y avons aussi associé la technique de similarité entre requêtes, celle-ci est fondée sur le principe que deux requêtes sont similaires si elles ont au moins un mot ou une partie d'un mot en commun, par exemple "Java Bean" et ".Java script " sont similaires parce qu'ils partagent le mot "Java".

Enfin, le prochain chapitre présentera l'implémentation de BLED

[2] Collaborative Filtering
[3] Web Usage Mining

Chapitre 6

Implémentation

Nous décrivons dans ce chapitre, l'implémentation de BLED. Nous allons commencer par une justification de choix des outils et de langages de programmation. Nous détaillons par la suite, le mode d'utilisation de toutes les fonctionnalités essentielles de BLED. Enfin, nous terminons ce chapitre en exposant des expériences concrètes.

6.1 Implémentation de BLED

BLED a été développé sous le système d'exploitation Linux (RedHat 9) et le serveur web Apache 2.0. Nous avons utilisé différents langages de programmation pour l'implémentation des différents modules. Par exemple, le module d'extraction des règles d'association a été implémenté en langage C++ et MySQL vu la nécessité de manipuler des structures de données complexes. Pour le reste des modules de système BLED, nous avons opté pour le PHP, MySQL et JavaS-

cript comme langages de programmation. Le choix de PHP se justifie par le fait qu'il est un logiciel libre, il peut fonctionner sous différents systèmes d'exploitation : Linux, Windows, Mac OS X, RISC OS. Il supporte également beaucoup de serveurs web tels que : Apache, IIS , PWS, OWP, etc. De plus, il est facile à apprendre et est capable de fonctionner en mode de commande, qui est très pratique pour réaliser des scripts qui s'exécutent régulièrement comme les crons daemon sous linux, en vue d'effectuer des tâches bien déterminées. En outre, PHP couvre presque tous les besoins d'un *webmaster*, il supporte également les protocoles de communications : POP3, IMAP, SNMP, LDAP, NNTP. Il gère des bases de données de type : MySQL, dBase, ODBC, PostGreSQL, Sybase, SQL et BD2 .

BLED est conçu pour fonctionner pour les versions 4 et plus d'Internet Explorer et Mozilla FireFox. Il fonctionne aussi sous Netscape sauf sous certaines versions où il peut y avoir des problèmes de compatibilité avec certaines fonctions écrites en JavaScript.

6.2 Interface de BLED

Dans cette section, nous allons présenter les différentes fonctionnalités de notre système BLED. La figure 6.1 présente l'interface de sa page web principale.

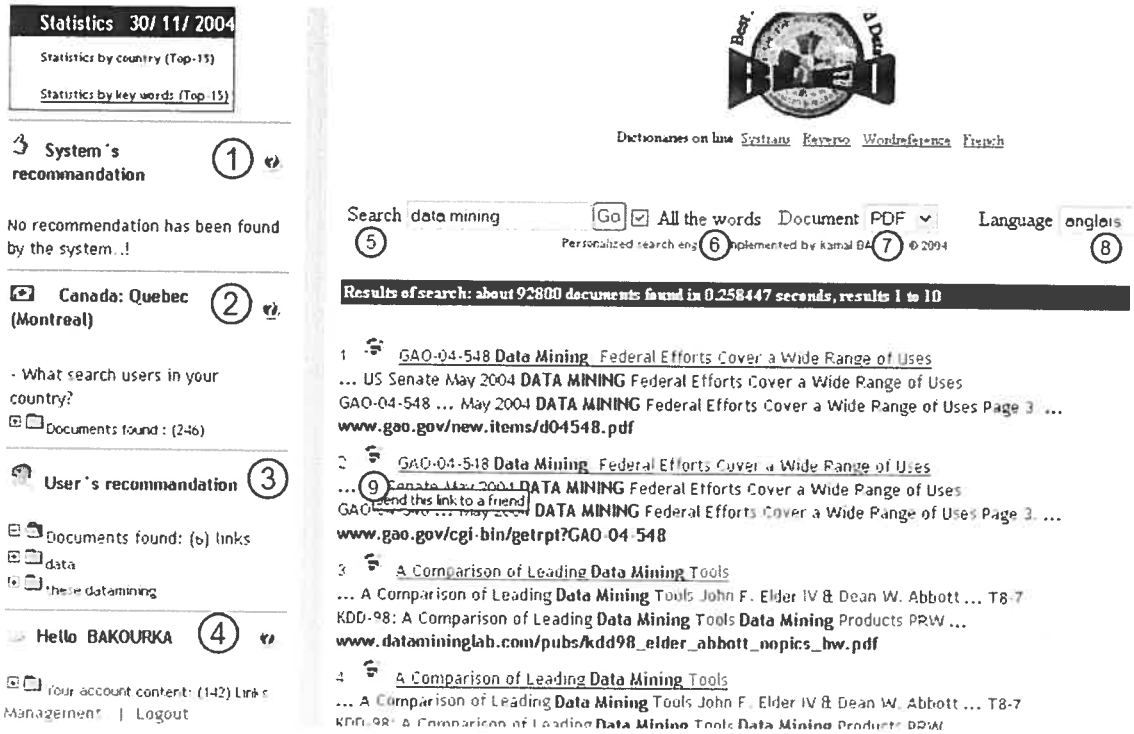

FIG. 6.1 – Page principale de BLED

6.2.1 Dossier personnel de l'usager

Cette fonctionnalité (voir figure 6.1 (4)) peut être considérée comme un support d'archivage, elle est disponible pour les utilisateurs ayant complété leur inscription dans le système, c'est-à-dire, ont créé leur compte dans BLED. Avec cette fonctionnalité, les utilisateurs peuvent garder dans leur dossier leurs recherches antérieures. Celles-ci contribuent énormément dans les trois autres fonctionnalités.

6.2.2 La recherche ciblée par pays

Cette fonctionnalité (voir figure 6.1 (2)) est remarquable, car elle permet de faire une recherche en se basant sur les tendances des utilisateurs par rapport à un pays. De plus, elle peut permettre aussi aux sites web commerciaux de cibler les produits trouvés en tenant compte des spécificités du pays. Par exemple, elle permettrait d'éviter de proposer des produits issus du Mexique alors que l'utilisateur, qui est aussi un potentiel acheteur, est au Canada. Notons que, plus les utilisateurs utilisent le système BLED plus celui-ci apprend et progresse et est capable d'offrir de meilleures recommandations. En ce sens, nous pouvons dire que BLED est un système qui s'enrichit et apprend par l'expérience.

6.2.3 Recommandation des utilisateurs

Cette fonctionnalité (voir figure 6.1 (3)) utilise l'évaluation faite par un utilisateur sur ses propres documents. En effet, chaque utilisateur a la possibilité d'évaluer le document qu'il a visité en lui attribuant une note entre 1 et 10, où 1 représente un document non pertinent et 10 un document très pertinent. Cette fonctionnalité peut être utilisée entre autres pour permettre aux utilisateurs inexpérimentés d'apprendre comment inférer les mots clés convenables afin d'affiner leurs recherches, ou encore pour recommander des documents évalués favorablement par un grand nombre d'utilisateurs. Elle se base sur une technique de corrélation entre mots clés recherchés qui s'appuie sur la base de données de BLED. C'est avec le langage d'interrogation de données MySQL que nous avons pu assurer cette fonctionnalité.

6.2.4 Recommandation du système

BLED utilise la technique connue sous le nom de règles d'association pour implémenter cette fonctionnalité (figure 6.1 (1)). Rappelons que cette technique sert à modéliser des phénomènes d'association entre objets, qui sont calculés dans notre cas à l'aide de l'algorithme APRIORI. Dans notre contexte, nous utilisons les règles d'association afin de recommander des documents pertinents en rapport avec une requête spécifique. Il faut noter que plus le volume de la base données de BLED est important, plus les règles d'association produites seront pertinentes et la recommandation sera meilleure.

6.3 Administration

L' `Administration`[1] permet de gérer un dossier personnel. En effet, l'usager peut voter les liens qu'il a jugé intéressants afin de les partager avec d'autres internautes. Il peut par contre supprimer de son dossier ceux qui ne le sont pas. Nous y trouvons (voir figure 6.2) les éléments suivants :

1. `Voter | Editer`[2] permet de voter ou modifier un vote déjà effectué sur un lien dans l'échelle de 1 à 10.
2. `Supprimer`[3] sert à supprimer un ou plusieurs liens de son dossier, les liens effacés disparaîtront complètement de son dossier.
3. `Zone de liste déroulante (Sélectionnez votre requête)` permet d'afficher tous les liens qui correspondent à une requête choisie.

[1] Management
[2] Rate| Edit
[3] Delete

4. `Zone de liste déroulante (Pages)` fixe le nombre de liens que l'utilisateur veut afficher par page : 10, 20, 30, etc.

5. `Date De`[4] `et date Au`[5] permet de localiser des recherches dans une période de temps.

6. `Début`[6] permet de se positionner sur la première page.

7. `Précédent`[7] permet de revenir à la page précédente.

8. `Suivant`[8] permet d'aller à la page suivante.

9. `Fin`[9] permet de passer directement à la dernière page.

10. `Case à cocher` permet la sélection d'un ou de plusieurs liens afin de les voter, les modifier ou les supprimer.

L'utilisateur peut trier le tableau de données selon des critères multiples : vote, date de visite, format du document et langue par un simple clic sur la colonne appropriée. Par exemple, si l'utilisateur souhaite trier ses documents selon le type de fichier (DOC, PDF, XLS, PPT) il n'a qu'à cliquer sur **"Format"** (voir figure 6.2). Notons qu'un lien doit être évalué supérieur ou égal à 5 pour qu'il apparaisse sur la rubrique dossier personnel de la page web principale de BLED.

[4] From
[5] To
[6] First
[7] Back
[8] Next
[9] End

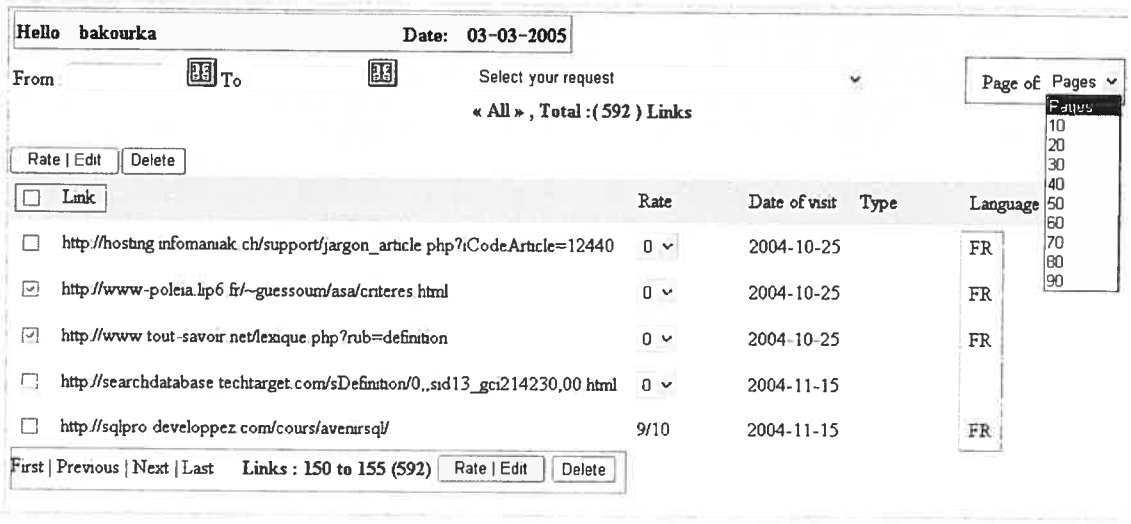

FIG. 6.2 – Gestion de dossier de l'usager

6.4 Inscription

Pour faire une inscription dans BLED, il suffit de remplir le formulaire d'inscription en passant par l'icône `devenir un membre`[10] ou à travers la rubrique `Inscrivez-vous ?`[11]. Les informations communiquées (voir figure 6.3) seront copiées dans une base de données des utilisateurs de BLED. Les informations qu'un nouvel utilisateur doit fournir sont :

- **Pseudonyme**[12] **:** Doit être composé d'au moins 4 caractères alphanumériques. Un message d'erreur s'affichera si ce pseudonyme a été pris par quelqu'un d'autre.
- **Mot de passe**[13] **:** Doit être composé d'au moins 4 caractères alphanumériques.
- **Adresse électronique**[14] **:** Sert à confirmer une inscription, elle pourrait aussi être utile au cas où l'utilisateur oubliera son mot de passe.

[10] become a member
[11] Subscribe ?
[12] username
[13] password
[14] email

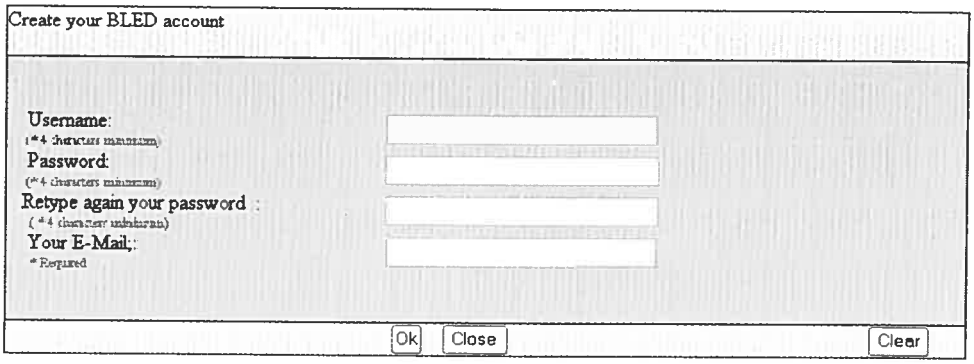

FIG. 6.3 – Devenir membre

6.5 Mot de passe oublié

Si l'utilisateur oublie son mot de passe, il suffit qu'il clique sur `mot de passe oublié`[15]. Il sera invité à saisir son **pseudonyme** et son **courrier électronique** qu'il a donné à l'inscription. Si ces informations sont correctes, il recevra un courrier électronique contenant son mot de passe (voir figure 6.4).

6.6 Identification

Cette rubrique ne peut être utilisée qu'après une inscription. Une fois que l'utilisateur est identifié, c'est-à-dire, après avoir introduit son **pseudonyme** et son **mot de passe**, BLED lui établira une session afin qu'il puisse garder trace de ses recherches, c'est-à-dire, ses historiques. Il lui permet également de gérer son dossier dans la section `Administrer`.

[15]password lost

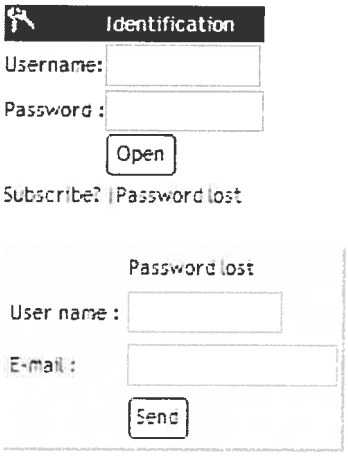

FIG. 6.4 – Gestion des utilisateurs

6.7 Recherche normale

Pour faire une recherche (voir figure 6.1), il suffit d'introduire un ou plusieurs mots clés dans la zone de saisie recherche[16] (voir figure 6.1 (5)), qui se trouve au milieu de l'écran de la page web principale de BLED en cliquant ensuite sur le bouton `Aller`[17] (selon bien sûr la langue choisie pour faire la recherche sur BLED).

Le résultat de cette recherche s'affiche sous forme de liste contenant les éléments suivants :

- Une icône permettant d'envoyer le document correspondant à d'autres utilisateurs (voir figure 6.1 (9)) ;
- Les liens complets pour accéder aux documents trouvés et leur descriptions ;
- Les noms des sites et leurs catégories.

Au dessus de cette liste s'affiche le nombre de références trouvées et la durée estimée de la recherche.

[16] Search
[17] Go

- Résultats de la recherche : Environ **2290000** documents trouvés en **0.274874** secondes. Il suffit de cliquer sur le bouton (⇒) pour passer à la page suivante et consulter les 10 liens suivants, et sur le bouton (⇐) pour revenir aux 10 liens précédents.

6.8 Recherche avancée

Il est possible de faire facilement des recherches avancées. Par exemple, pour effectuer une recherche sur un groupe de mots dans un ordre fixe, il suffit de cocher la case `Tous les mots`[18] (voir figure 6.1 (6)). Si l'utilisateur souhaite rechercher les liens dans lesquels figure le concept **festival de Jazz à Montréal**, seules les références contenant cette expression seront affichées. Il est également possible de préciser le type de document (voir figure 6.1 (7)) (pdf, ppt, doc, etc.), la langue (voir figure 6.1 (8)) (Français, Anglais,etc) ou d'utiliser les opérateurs : (+), (-), (~), (*) afin de mieux approfondir une recherche.

1. L'opérateur (~)

Lorsque cet opérateur est fixé devant un mot, dans une expression, il incite BLED à trouver tous les documents comportant ce mot, ainsi que les synonymes qui en dépendent. Par exemple, si un utilisateur recherche l'expression *"~car"*, BLED lui suggère les documents contenant des mots comme : *"automobile", "car", "vehicle", "bus" etc.*

2. L'opérateur (-)

Lorsque l'opérateur (-) est fixé devant un mot dans une expression, BLED comprend qu'il va falloir exclure tous les documents (liens) comportant ce mot. Par

[18] All words

exemple, pour chercher des informations concernant les virus non informatique, il suffit de taper l'expression *"virus -computer"*, BLED éliminera tous les documents qui contiennent le mot *"computer"*.

3. L'opérateur (+)

Lorsque cet opérateur est placé devant un mot, dans une expression, il exige la présence de ce mot dans tous les documents trouvés. Par exemple, lorsque l'expression *"programmation +web"* est recherchée, tous les documents proposés par BLED doivent contenir le mot web.

4. Le joker (*)

Tout comme dans les expressions régulières, lorsque le symbole (*) paraît dans une expression, cela indique que cette expression peut contenir n'importe quelle autre expression à la place de (*). Par exemple, si l'expression *"nous * capables"* est recherchée, BLED peut recommander des documents comportant des expressions comme : *"Sommes-nous **vraiment** capables de choisir ?, Nous **sommes** capables de vivre avec une telle entente* ou *Nous **sommes** capables de tenir l'Iran, etc."*

6.9 Expérimentation

La première étape de notre démarche expérimentale consistait à déterminer tout au long d'une période de 6 mois (entre Juillet 2004 et Décembre 2004) toutes les données nécessaires au traitement. Cette durée a été considérée suffisante pour accomplir et réaliser la tâche fondamentale de notre recherche, qui est la recommandation automatique de documents/ services en utilisant la technique des règles d'association.

En ce qui concerne les autres fonctions de BLED : la recommandation automatique des usagers, le dossier personnel de l'usager et la recherche par pays ont

été réalisées avec succès, car elles n'utilisent pas des algorithmes compliqués en forage de données, mais des algorithmes composés de requêtes établies en MySQL et de codes PHP.

L'analyse de la base de données principale de BLED a donné les résultats que nous allons montrer dans les paragraphes qui suivent.

6.9.1 Les utilisateurs potentiels

Les lignes de code MySQL ci-après permettent de connaître la distribution totale des requêtes demeurant dans la base de données BLED et provenant de différents utilisateurs.

```
SELECT id, count(id) AS top
FROM data
GROUP BY id ORDER BY top DESC
```

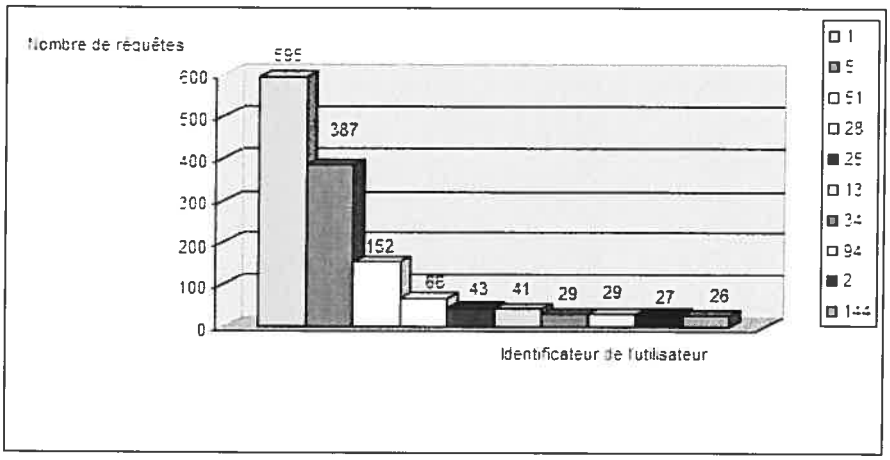

FIG. 6.5 – Le top 10 des utilisateurs les plus actifs

D'après le graphique 6.5, nous avons constaté que cette distribution n'a pas été tout à fait équilibrée entre les utilisateurs. En effet, la base de données BLED

contient 2024 enregistrements qui sont répartis sur 114 utilisateurs, dont 29.39% de ces enregistrements appartiennent à l'utilisateur numéro (1), 19.12% appartiennent à l'utilisateur numéro (5) et 7.5% appartiennent à l'utilisateur numéro (51). Ces trois utilisateurs constituent 56.01% de la base de données de BLED, la part 43.99% était évidement partagée entre le reste des utilisateurs, cela a affecté le processus de l'extraction des règles d'association, car il diminuait le nombre de transactions.

6.9.2 L'utilisation de BLED

Nous avons ainsi interrogé la base de donnée de BLED afin de connaître respectivement la portée de l'utilisation de notre moteur de recherche, sur l'échelle internationale, (figure 6.6) pendant les 30 semaines de Juillet 2004 jusqu'à Décembre 2004 (figure 6.7).

```
SELECT pays, COUNT(id) FROM data
GROUP BY pays
```

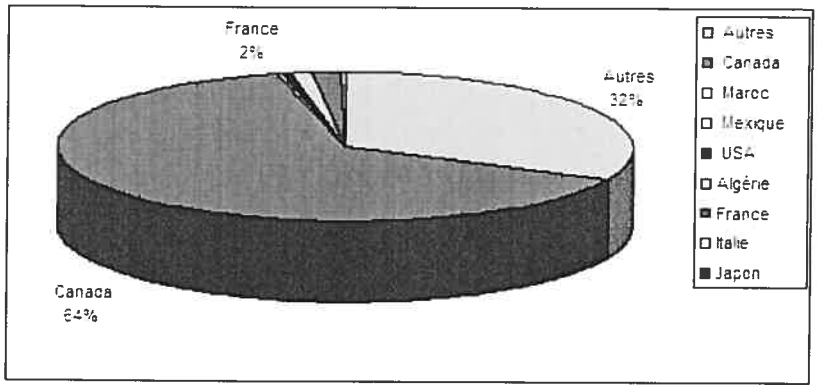

FIG. 6.6 – BLED

La figure 6.6 nous indique que la majorité des recherches faites sur BLED étaient localisées au CANADA (64%). D'autres étaient constatées en France (2%), en Algérie, etc.

```
SELECT  WEEK(save) AS semaine, COUNT(id) FROM data GROUP BY
semaine;
```

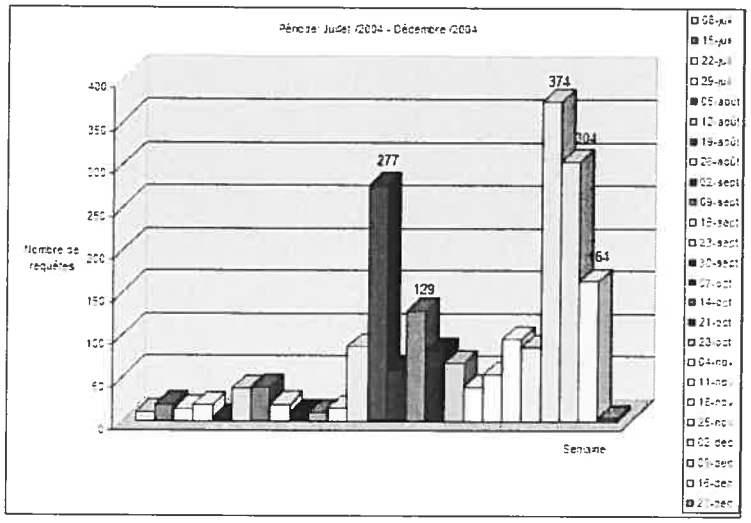

FIG. 6.7 – Fréquence d'utilisation de BLED par semaine

La figure 6.7 montre l'augmentation notable de la taille de la base de données de BLED, entre 02 décembre 2004 et le 15 décembre. En effet, pendant cette période nous avons pu récolter environ 41,46% de données. Cela est dû au fait que nous étions dans l'étape finale de la réalisation de BLED. Alors nous avons demandé à nos collègues de l'université de faire des recherches libres sur BLED.

6.9.3 Les requêtes les plus recherchées

Le code MySQL présenté ci-après permet d'interroger la base de données BLED afin d'en déduire quelles sont les requêtes les plus recherchées pendant

la période de Juillet 2004 et Décembre 2004. Cette opération est consistante en terme de révélation des tendances des internautes.

```
SELECT request, count(id) AS top FROM \textbf{data}
GROUP BY request ORDER BY top DESC LIMIT 10
```

TAB. 6.1 – Le TOP 10 des requêtes les plus recherchées

N°	Requêtes	Recherchées
1	salle de théâtre	45
2	sept merveilles du monde	37
3	tourisme	31
4	programmation java	30
5	houfani	26
6	quotient intellectuel	24
7	egypte	20
8	humour	19
9	université	18
10	droit de l'homme	18

6.9.4 Génération des règles d'association

Dans cette section nous montrons avec des données réelles comment retrouver les règles en utilisant l'algorithme APRIORI (voir section 5.2). Rappelons que cette opération nécessite les étapes suivantes : 1°) **la sélection et préparation des données** consiste à établir la liste de tous les documents appréciés par les utilisateurs de BLED selon les critères que nous avons imposé par notre étude, 2°) **l'identification des transactions des utilisateurs** permet d'extraire toutes les transactions faites par les utilisateurs depuis juillet 2004 jusqu'à décembre 2004 en utilisant pour cela la base de données principale de BLED et la liste récupérée par l'opération précédente, notons qu'une transaction d'un utilisateur représente tous les documents qu'il vote supérieur ou égal à 5, à condition que, ces documents soient présents dans la liste établie par l'étape de la sélection et

préparation de données, 3°) **les règles d'association** consiste à opérer l'algorithme APRIORI sur cette base de transactions afin de générer toutes les règles d'association possibles qui seront utilisées à des fins de recommandation.

6.9.4.1 Sélection et préparation de données

La portion du code présentée ci-dessous permet de sélectionner tous les documents votés au moins par une personne et dont la moyenne est supérieure ou égale à 5 en attribuant un identifiant unique à chaque document ainsi obtenu. Le tableau 6.2 présente un extrait de ces documents.

```
SELECT url, COUNT(id), AVG(vote) AS Moyenne FROM data
WHERE vote > 0 GROUP BY url
HAVING AVG(vote)>= 5 AND COUNT(id) >= 1
ORDER BY Moyenne DESC;
```

TAB. 6.2 – Extrait des URL appréciés

Code URL	URL	Voté par	Moy.
1	http ://www.meilleursprix.ca/	1	10.00
2	http ://www.bittorent.biz	1	10.00
.	.	.	
.	.	.	
.	.	.	
70	http ://www.100cv.com/	2	8.00
.	.	.	
.	.	.	
.	.	.	
386	http ://q.cis.uoguelph.ca/~skremer/	2	5.00

6.9.4.2 Identification des transactions des utilisateurs

Nous avons constaté dans cette étape de sélection et préparation de données que 76.77% des enregistrements ont été éliminés de la base de données soit (1554

enregistrements), cela revient aux critères que nous avons fixé par notre étude, c'est-à-dire, nous gardons seulement les documents votés au mois par un utilisateur à une moyenne supérieure ou égale à 5. Car nous avons estimé que pour qu'un document (URL) soit pertinent il faudrait qu'il soit apprécié par la majorité des utilisateurs qui l'ont voté. Pour des raisons inconnues, il arrive parfois qu'un utilisateur a vraiment apprécié un document, mais il ne l'a pas voté, le système alors ne prend pas en considération ce document pour calculer la moyenne. Par exemple, supposons qu'un document "URL" est visité par 4 utilisateurs, dont 3 utilisateurs l'ont voté respectivement à : 5, 4 et 7, le quatrième utilisateur ne l'a pas voté, alors la moyenne qui sera affectée à ce document est 5.33. Cependant, si nous comptons l'utilisateur 4, cette moyenne devient 4.
Il est à noter que ces critères d'évaluation des documents sont subjectifs et non objectifs et dépendent de la façon dont les utilisateurs les apprécient.
La liste globale de toutes les transactions est citée en **annexe A**.

6.9.4.3 Règles d'association

Nous avons appliqué l'algorithme APRIORI sur la matrice des transactions $\mathcal{A}(30, 386)$ comprenant **30 transactions** (Utilisateurs) et **386 items** (URLs) :

$$\mathcal{A}(30,386) = \begin{pmatrix} 1 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Où

$$a_{ij} = \begin{cases} 1 & \text{si l'utilisateur i a voté l'URL}_j \geq 5 \\ 0 & \textit{sinon} \end{cases}$$

Le tableau 6.3 résume le nombre de règles d'association obtenu en fonction des paramètres $minsupport$ et $minconfiance$.

TAB. 6.3 – Règles d'association avec APRIORI

Support	Confiance	Nombre de règles d'association
10%	50%	0
10%	75%	0
10%	100%	0
6%	50%	2228736
6%	66%	2228736
6%	100%	2228736
3%	50%	Pas de réponse
3%	100%	Pas de réponse

Le tableau 6.3 montre que lorsque le nombre d'items (URLs) à gérer est important l'algorithme APRIORI ne converge pas. En effet, pour générer les $(100-itemsets)$ nous devons générer tous les candidats possibles, c'est-à-dire, $2^{100} - 1 \simeq 10^{30}$ candidats [44]. Alors, pour traiter 386 items avec APRIORI, il est absolument nécessaire d'utiliser une machine puissante, en terme de mémoire et de processeur.

Toutefois, notre objectif est de générer des règles d'association et de l'exploiter pour des fins de recommendations et non pas de trouver des remèdes aux problèmes de l'algorithme APRIORI. C'est pourquoi, nous suggérons l'utilisation de l'Analyse Formelle de Concepts (AFC)[19] [31], qui consiste d'abord à trouver les

[19]Formal Concept Analysis

itemsets fermés fréquents[20][60], ensuite de générer les règles d'association à partir de ces itemsets fermés fréquents. De plus, il a été montré que cette approche est très efficace, car elle permet de générer des règles informatives, c'est-à-dire, les règles ayant une prémisse minimale et une conséquence maximale, ceci évitera d'avoir des règles redondantes. C'est pourquoi nous avons utilisé le logiciel Galicia[21] (**GA**lois **L**attice-based **I**ncremental **C**losed **I**temset **A**pproach). L'avantage de celui-ci, est qu'il permet la visualisation des règles d'association et les itemsets fermés fréquents sous forme de treillis.

Nous avons alors opéré GALICIA sur la même liste de transactions de **l'annexe A** et nous avons obtenu les résultats du tableau suivant :

TAB. 6.4 – Règles d'association avec GALICIA

Support	Confiance	Nombre de règles d'association
10%	50%	1
10%	75%	2
10%	100%	7
6%	50%	2
6%	66%	20
6%	100%	67
3%	50%	158
3%	100%	609

6.10 Conclusion

Dans ce chapitre, nous avons exposé en détail le mode de fonctionnement de BLED, ces différentes fonctionnalités telles que : le dossier personnel de l'usager, la recommandation des utilisateurs fondée sur la similarité entre les requêtes, la recommandation du système basée sur les règles d'association, etc.

[20] Frequent Closed Itemsets

[21] http ://www.iro.umontreal.ca/~galicia/

Bien que les recommandations fournies par BLED soient bonnes, elles auraient été meilleures si nous avions davantage d'informations. En effet, la base de données de BLED ne contient pour l'instant que 2048 documents.

En revanche, le problème des treillis que nous avons rencontré, en utilisant l'algorithme APRIORI, pendant la phase de génération des règles d'association, a été résolu avec succès grâce au logiciel Galicia.

Le prochain chapitre résumera notre solution en exposant ses forces et ses faiblesses.

Chapitre 7

Discussion et conclusion

Ce chapitre clôture ce mémoire en discutant les forces et les faiblesses du système BLED, en effectuant une comparaison avec d'autres systèmes plus récents. Il expose également comme travaux futurs des solutions pouvant améliorer grandement l'efficacité de BLED.

7.1 Discussion

Nous avons présenté dans ce mémoire une approche d'aide pour la recherche d'informations personnalisées sur Internet. Nous sommes conscients que nous ne disposons pas encore d'assez de données (utilisateurs, requêtes, documents visitées, etc.) pour illustrer d'avantage l'efficacité du système. Toutefois, nous pensons que l'étude menée ici est un premier pas dans cette direction.

7.2 Forces

L'avantage de notre approche BLED réside non seulement dans l'utilisation des résultats renvoyés en interrogeant la base de données de Google, mais également dans l'évaluation de ces documents par d'autres utilisateurs. Ce qui donne des résultats plus affinés sous formes de pyramides. En effet, avec BLED les utilisateurs pourraient économiser le temps consacré à la recherche d'informations sur Internet en profitant des recherches effectuées par d'autres utilisateurs. Ils peuvent également apprendre comment élaborer leurs requêtes en visant précisément les informations qu'ils cherchent. En conséquence, les utilisateurs jouent un rôle très important pour améliorer la performance de BLED et enrichir les recommendations qu'il fournit. Pour ceux qui voudraient le tester, BLED est disponible en deux versions anglaise et française à l'adresse suivante :
http ://www-etud.iro.umontreal.ca/~bakourka.
Afin de mettre en évidence l'efficacité et la valeur de notre moteur de recherche BLED, nous l'avons comparé à d'autres systèmes récents.
Voici quelques critères de comparaison en rapport avec la personnalisation de la recherche d'informations sur Internet que nous avons utilisé (voir tableau 7.1) :

1. Historique

Le système garde-t-il une trace des actions de l'utilisateur et lui permet-t-il un accès à son propre dossier ?

2. Recommandation

Le système peut-il recommander des documents attractifs aux autres utilisateurs, soit automatiquement par le biais du moteur de recommandation du système, soit manuellement en spécifiant l'adresse du courrier électronique de la personne destinataire ?

3. Evaluation

Le système permet-il aux utilisateurs d'évaluer les liens visités ? Et utilise-t-il ces informations pour effectuer sa recommendation ?

4. Feedback

Le système permet-t-il à l'usager de bloquer les URLs qu'il ne souhaiterait pas revoir lors des prochaines recherches ?

7.2.1 Description des systèmes

Le paragraphe suivant présente brièvement les systèmes que nous souhaitons comparer avec BLED.

1. FOOXX[1]

Paru sous une version française en juillet 2004, ce système utilise les informations recueillies auprès des utilisateurs pour déterminer la pertinence des pages web. Il offre plusieurs manières pour recommander des pages intéressantes à d'autres utilisateurs. De plus, il permet aux utilisateurs de bloquer les liens qu'ils jugent inutiles.

2. A9[2]

A9 c'est un moteur de recherche puissant crée par Amazon alimenté de plusieurs ressources. A titre d'exemple, pour des recherches visant des documents web ou des images A9 utilise Google, pour la recherche de livres il utilise Amazon et pour la recherche de films il utilise IMDB. A9 permet d'annoter n'importe quel type de document web, de stocker et d'organiser des signets. Il utilise aussi l'historique de l'utilisateur pour lui recommander des sites web similaires.

[1] http ://www.fooxx.com

[2] http ://www.a9.com

3. Ask Jeevs[3]

Une nouvelle fonctionnalité appelée "My Jeeves" inventée par la compagnie Ask est parue en septembre 2004. Cette fonctionnalité permet d'établir des recherches personnelles, les organiser, les partager avec d'autres personnes, les imprimer et ajouter même des notes à ces recherches. Le moteur de recherche Ask Jeevs utilise deux méthodes différentes pour accomplir ces tâches. La première solution consiste à utiliser les cookies, dont nous avons présenté les inconvénients dans la section 4.1.1. La deuxième méthode consiste à employer l'authentification par un pseudonyme et un mot de passe. C'est la solution que nous avons choisie pour notre approche, car jusqu'à présent c'est le seul moyen qui permet d'identifer une personne.

TAB. 7.1 – Comparaison de BLED à d'autres systèmes

Fonctionnalités	Historique	Recommandation	Évaluation	FeedBack
http ://www.a9.com/	Oui	Oui	Non	Non
http ://myjeeves.ask.com/	Oui	Oui	Non	Non
http ://www.fooxx.com/	Oui	Oui	Non	Oui
BLED	Oui	Oui	Oui	Non

A travers ce tableau, nous remarquons que notre système BLED remplit presque tous les critères, qui sont généralement attendus par les internautes lors de leurs recherches tels que l'historique, la recommandation et l'évaluation. Malheureusement, nous n'avons pas encore pris en compte le critère de "FeedBack" que le moteur de recherche Fooxx.com utilise. Ce critère est un élément important dans la personnalisation des moteurs de recherche. Nous pensons toutefois l'intégrer au système BLED dans un avenir proche afin d'améliorer ses performances.

[3]http ://myjeeves.ask.com

7.3 Faiblesses

Nous avons dit précédemment que notre solution depend potentiellement de la taille de la base de données opérationnelle. Et plus le volume de cette base de données croît plus la recommendation de documents devient robuste. Cela peut se justifier par le fait que la technique des règles d'associations nécessite un énorme jeu de données pour générer des règles d'associations plus efficaces.
De même, la non homogénéité de la structure des documents présents sur le web rend difficile (voire impossible) de concevoir un système qui répond aux exigences et aux besoins de ses utilisateurs.

7.4 Perspectives

Comme travaux futurs, nous pensons améliorer les performances de notre système en utilisant des techniques de catégorisation automatique de documents afin d'organiser les documents en catégories, comme les annuaires de références. Notre base de données actuelle stocke toutes les informations nécessaires en prévision de cette tâche future. Certaines études ont proposé des approches simples permettant la segmentation des requêtes des utilisateurs, en utilisant les traces laissées par ceux-ci lors de leurs navigations [76, 77]. L'un des principes de ces approches est le suivant : si plusieurs requêtes pointent sur un même document alors ces requêtes sont considérées semblables [39]. Cette idée pourrait améliorer grandement notre approche basée sur la similarité entre mots clés des requêtes.

Annexe A

Ci-dessous, la liste des transactions des utilisateurs que nous avons obtenues en utilisant MySQL. Le code suivant permet de sélectionner tous les documents votés au moins par une personne et dont la moyenne est supérieure ou égale à 5/10.

```
SELECT url, COUNT(id), AVG(vote) AS Moyenne
FROM data WHERE vote > 0 GROUP BY url HAVING AVG(vote)>= 5 AND
COUNT(id) >= 1 ORDER BY Moyenne DESC;
```

Notons qu'une transaction est constituée d'un ensemble des URLs votés supérieurs ou égaux à 5/10 par un seul utilisateur. Par exemple, l'utilisateur T01 a apprécié les URLs identifiés par les code : 53 126 142 154 156 158 180 230 240 242 244 280 288 322 331 334 374 378 et c'est pour cette raison qu'il les a voté supérieur ou égal à 5/10.

A.1 Liste des transactions

- **T00** : 1 2 3 4 5 7 11 19 23 24 28 30 37 38 41 49 50 51 52 54 55 57 60 65 66 71 74 84 90 91 93 94 96 97 98 99 105 106 107 109 111 113 115 116 118 119 122 123 125 130 132 135 137 145 155 160 162 163 171 172 174 178 183 185 186 188 193 206 209 212 215 218 222 224 226 229 233 235 236 249 253 254 256 257 258 260 261 262 264 265 268 270 271 272 274 275 276 278 279 282 283 284 285 286 287 291 292 293 295 296 297 298 299 301 302 303 304 305 306 307 308 309 310 312 314 316 317 319 320 321 325 326 327 328 329 330 332 333 335 336 338 339 340 341 342 343 345 348 349 350 353 354 357 360 363 364 366 368 369 370 371 372 373 376 377 379 381 382 384 386
- **T01** : 53 126 142 154 156 158 180 230 240 242 244 280 288 322 331 334 374 378
- **T02** : 117 136 147 161 191 220 243 259 344 346
- **T03** : 52 90 134 136 155 158 163 182 197 224 259 263 266 355
- **T04** : 13 17 27 29 34 36 43 44 80 82 83 85 133 153 176 203 208 290
- **T05** : 104 140 146 190 223 375
- **T06** : 87 97 114 136 143 194 199 232 248 252 351
- **T07** : 195
- **T08** : 15 16 31 32 42 45 64 73 91 92 147 148 150 152 157 216 241
- **T09** : 170 243 246 247 318 323 358 361
- **T10** : 115 142 166 201 244 246 247 358 361 365
- **T11** : 210
- **T12** : 21 35 46 48 53 76 129 204 251
- **T13** : 20 64 67 75 76 86 87 89 95 110 141 143 152 177 182 211 237 242 245 251 300 337
- **T14** : 63 68 70 79 81 89 95 102 120 121 136 139 146 148 149 151 153 157 158 173 180 190 216 231 240 241 252 281
- **T15** : 205 289 380

- **T16** : 39 56 92 94 108 124 127 136 144 181 192 213 221 239 294 347 385
- **T17** : 147
- **T18** : 53 59 68 70 88 89 95 102 128 136 138 146 147 148 149 151 154 156 157 158 159 165 169 173 179 180 184 187 196 198 199 200 202 207 214 225 228 230 231 234 238 240 241 245 248 252 255 352
- **T10** : 167 267 269 273 311 313 315 330 356 359 362 367
- **T20** : 89 97 146 189 219 277
- **T21** : 324
- **T22** : 91 150 175
- **T23** : 251
- **T24** : 131
- **T25** : 62 100 117
- **T26** : 72
- **T27** : 227
- **T28** : 47 69 78
- **T29** : 217

Annexe B

B.0.1 MSN Search

MSN Search[1] est le moteur de recherche utilisé par le portail MSN de Microsoft. Celui-ci utilise trois bases de données différentes : la base de données d'Inktomi pour son moteur de recherche, la base de données de LookSmart pour son annuaire et la base de données d'Overture pour les sites sponsorisés. Un de ses avantages est qu'il ne recherche pas les mots clés susceptibles d'afficher du contenu réservé aux adultes [69, 68, 11].

B.0.2 Google

Google[2] est devenu le moteur de recherche le plus remarquable. En février 1999, il s'est transformé de la version alpha à la version bêta. Il a été officiellement lancé le 21 septembre 1999. En juin 2000 Google a annoncé une base de données d'environ 560 millions pages web, et vers la fin de l'an 2000 ce chiffre a augmenté

[1] www.msnsearch.com

[2] www.google.com

jusqu'a 600 millions, puis 1.5 milliards pages web en 2001, par la suite 2 milliards pages web sur sa page web principale. En novembre 2002, il a réclamé jusqu'à 3 milliards de pages web, en février 2004 ce chiffre a atteint les 4 milliards. En décembre 2004, Google proclame un chiffre qui devance les 8 milliards de pages web [34] sur sa page web principale. Ceci montre la progression rapide de l'index de ce moteur de recherche [16, 69, 68, 11].

B.0.3 Teoma

Téoma[3] est un moteur de recherche paru au printemps 2001. Il construit sa propre base de données. Cette base contient également les liens promotionnels provenant de la base de données d'AdWords de Google. Son avantage se montre en l'identification des méta-sites et le raffinement des résultats de recherche qui se basent sur les communautés du web. Il est à noter que Teoma a été acheté en septembre 2001 par Ask Jeeves Inc. [69, 68, 11].

B.0.4 Altavista

AltaVista[4] qui dénote *"vue d'en haut"*, a été créé en 1995 par des scientifiques du laboratoire de recherche en informatique de Palo Alto, en Californie. Son objectif principal était de développer une base de données de recherche de texte intégral sur la toile WWW. En février 2003, altavista a été acheté par Overture. Celle-ci a envisagé en 2003 de fusionner la base de données d'Altavista et celle d'AllTheWeb . Cependant, la base de données d'Altavista a été remplacée par

[3] www.teoma.com

[4] www.altavista.com

celle de Yahoo! /Inktomi le 25 mars 2004 car Overture a été achetée par Yahoo!. Depuis ce jour, Altavista n'utilise plus sa base de données, mais une sous-base de celle utilisée par Yahoo!. Ce qui est intéressant dans ce moteur de recherche est sa fonction originale offerte pour la recherche de documents audio et vidéo. Elle est paramétrable en fonction des principaux formats de fichiers mp3, wma, etc., la durée de ces fichiers et leur provenance. Le traducteur en ligne d'Altavista permet de traduire du texte ou une page web en huit langues différentes [69, 68, 11].

B.0.5 DMoz

Open Directory Project[5] connu aussi sous le nom de NewHoo. Il a été fondé en 1998 dans l'esprit du mouvement Open Source. Il est la propriété de AOL Time Warner. DMoz est totalement gratuit pour y soumettre un site web. En avril 2004 DMoz a recensé presque 4.575.000 pages web réparties en 590.000 catégories. À nos jours, DMoz est très utilisé par la majorité des moteurs de recherche comme AOL Search, HotBot, Google, Lycos, Ask Jeeves, Altavista, Netscape [69, 68, 11].

B.0.6 Yahoo!

Yahoo [6] est paru en 1993. Ses fondateurs sont Jerry Yang et David Filo. La technologie mise en oeuvre baptisée Yahoo! Search Technology lui a permis de construire et d'établir ses différentes bases de données selon les nécessités. Par exemple, la base de données contenant les pages web qui découlent de celle

[5] www.dmoz.org

[6] www.yahoo.com

d'Inktomi, les liens promotionnels proviennent de la base de données d'Overture. Yahoo! détient également les pages jaunes, les news, etc. [69, 68, 11]

B.0.7 Ask Jeeves

Ask Jeeves [7] a été réalisé à Berkeley, en Californie en 1995. Comme son nom l'indique **ask** se traduit par **"demander"**. Ask Jeeves possède une fonction très originale qui permet de fournir des réponses à des questions formulées en langage naturel. Par exemple, il répond à la question "***what is search engine ?***" par "***search engine : a computer program that retrieves documents or files or data from a database or from a computer network (especially from the Internet)***". Cet outil se définit comme étant un système "Question-Réponse". En revanche, cette fonction n'est disponible qu'en version anglaise. Ce moteur de recherche fournit en outre des réponses complémentaires provenant du moteur de recherche Teoma dont il est le propriétaire [69, 68, 11].

Ask Jeeves a récemment introduit une nouvelle fonction sous une version bêta dont nous ne connaissons pas la technologie implémentée. Elle est dédiée spécifiquement à la personnalisation de la recherche d'informations sous la rubrique *My AskJeeves*. Elle est disponible sur le site : Myjeeves.ask.com.

[7] www.ask.com

La figure suivante montre l'interaction existant entre les moteurs de recherche les plus connus [43].

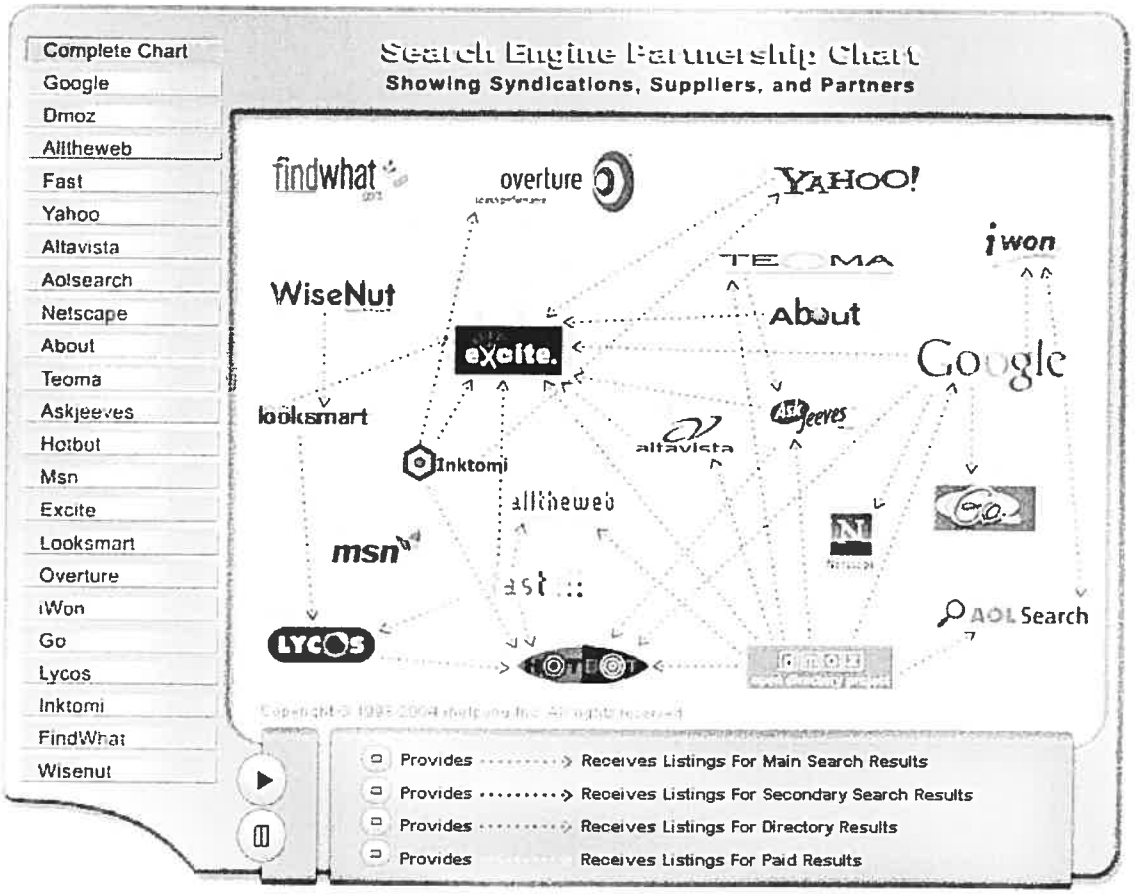

FIG. B.1 – Interaction entre les moteurs de recherche

Bibliographie

[1] Aamodt A., Plaza, E. "Case-based reasoning : foundational issues, methodological variations, and system approaches". *AI Commun.*, 7(1) :pages 39–59, 1994.

[2] Adamo, J. *Data Mining for Association Rules and Sequential Patterns.* Springer, 2001.

[3] Agrawal, R., Imielinski, T., Swami, A. "Mining Association Rules between Sets of Items in Large Databases". In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.

[4] Agrawal, R., Srikant, R. "Fast Algorithms for Mining Association Rules". pages 487–499, 1994.

[5] Almeida, V., Bestavros, A., Crovella, M., Oliveira, A. "Characterizing reference locality in the www". 1996.

[6] Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S. "Searching the Web". *ACM Transactions on Internet Technology*, 2001.

[7] Armstrong, R., Freitag, D., Joachims, T., Mitchell, T. "Webwatcher : A learning apprentice for the World Wide Web". *Proceedings of the AAAI Spring Symposium on Information Gathering*, pages 06–12, 1995.

[8] Arocena, G. , Mendelzon, A. "WebOQL : Restructuring Documents, Databases and Webs". pages 24–33, 1998.

[9] Balfe, E., Smyth, B. "Case-based collaborative Web search". *Proceedings of the European Conference on Case-Based Reasoning (ECBR'04)*, pages 489–503, 2004.

[10] Balfe, E., Smyth, B. "Query Mining for Community Based Web Search". pages pages 594–598, 2004.

[11] BCI. "BRUCE CLAY, INC". URL : http ://www.bruceclay.com/, Janvier 2005.

[12] Becker, S. *Data Warehousing Web Engineering.* IRM Press, 2002.

[13] Berkhin, P. "Survey Of Clustering Data Mining Techniques". Technical report, Accrue Software, San Jose, CA, 2002.

[14] Berry,M., Linoff, G. *"Data Mining Techniques : For Marketing, Sales, and Customer Relationship Management".* Wiley Computer Publishing; 2 edition, 2004.

[15] Bollacker, K., Lawrence, S., Giles, C. "CiteSeer : An autonomous web agent for automatic retrieval and identification of interesting publications". *Proceedings of the 2nd International Conference on Autonomous Agents (AA'98)*, pages 116–123, 1998.

[16] Brin, S., Page, L. "The anatomy of a large-scale hypertextual Web search engine". *Computer Networks and ISDN Systems*, 30(1-7) :pages 107–117, 1998.

[17] Buchner, A., Mulvenna, M. "Discovering Internet Marketing Intellegence Through Online Analytical Web Usage Mining". *SIGMOD Record*, 27(4) :pages 54–61, 1998.

[18] Cadez, I., David, H., Meek, C, Smyth, P., White, S. "Model-Based Clustering and Visualization of Navigation Patterns on a Web Site". pages 280–284, 2000.

[19] Castillo, C., Marin, M., Rodriguez, A., Baeza-Yates, R. "Scheduling algorithms for Web crawling", 2004.

[20] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., Rajagopalan, S. "Automatic resource list compilation by analyzing hyperlink structure and associated text". In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[21] Chau, M., Chen, H. "Personalized and Focused Web Spiders", Février 2003.

[22] Church, K., Keane, M., Smyth, B. "The First Click is the Deepest : Assessing Information Scent Predictions for a Personalized Search Engine". In *Proceedings of the 3rd Workshop on Empirical Evaluation of Adaptive Systems. 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. Eindhoven, The Netherlands.*, 2004.

[23] Cooley, R., Mobasher, B., Srivastava, J. "Data Preparation for Mining World Wide Web Browsing Patterns". *Knowledge and Information Systems*, 1(1) :pages 5–32, 1999.

[24] Cooley, R., Srivastava, J., Mobasher, B. "Web Mining : Information and Pattern Discovery on the World Wide Web", 1997.

[25] Craven, P. "Google PageRank Explained". URL : http ://www.webworkshop.net/pagerank.html, Février 2005.

[26] Eirinaki, M., Vazirgiannis, M. "Web mining for web personalization". *ACM Trans. Inter. Tech.*, 3(1) :pages 01–27, 2003.

[27] El-Hajj, M., Zaïane, O. "Parallel Association Rule Mining With Minimum Inter-Processor Communication". In *Proceedings of the 3rd ACM internatio-*

nal workshop on Data warehousing and OLAP, pages 519–523. IEEE Computer Society, 2003.

[28] Frias-Martinez, E., Karamcheti, V. "A Prediction Model for User Access Sequences". *"Proceedings of the WEBKDD Workshop : Web Mining for Usage Patterns and User Profiles, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining"*, Juillet 2002.

[29] Fu, X., Budzik, J., Hammond, K. "Mining navigation history for recommendation". *Intelligent User Interfaces*, pages 106–112, 2000.

[30] Galeas, P. "Web Mining". URL : http ://www.galeas.de/webmining.html, Janvier 2005.

[31] Ganter, B., Wille, R. *"Formal Concept Analysis, Mathematical Foundations"*. Springer, Berlin, 1999.

[32] Gery, M., Haddad, H. "Evaluation of web usage mining approaches for user's next request prediction". In *Proceedings of the fifth ACM international workshop on Web information and data management*, pages 74–81. ACM Press, 2003.

[33] Goglin, J. *La construction du datawarehouse : du datamart au dataweb.* Hermis, 1998.

[34] Google. "Page principale". URL : http ://www.google.com, Janvier 2005.

[35] Groth, R. *"Data Mining Building competitive Advantage"*. Prentice Hall PTR, 2000.

[36] Hammond, K., Burke, R., Martin, C., Lytinen, S. "Faq-finder : A case-based approach to knowledge navigation". 1995.

[37] Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J. "WebACE : A web agent for document cate-

gorization and exploration". *Proceeding of the 2nd International Conference on Autonomous Agents (AA'98)*, pages p408–415, 1998.

[38] Han, J. ,Kamber, M. *"Data Mining : Concepts and Techniques"*. Morgan Kaufmann, 2002.

[39] He, B. , Tao, T., Chang, K. "Clustering StructuredWeb Sources : a Schema-based, Model-Differentiation Approach". In *In Proceedings of the EBDT Workshop on Clustering Information over the Web (EDBT-ClustWeb'04)*, volume 3268, Mars 2004.

[40] Hu, W. "An overview of the World Wide Web search technologies", 2001.

[41] Hu, W., Yeh, J. "World Wide Web Search Technologies", 2003.

[42] Huberman, B., Pirolli, P., Pitkow, J., Kukose, R. "Strong regularities in World Wide Web Surfing". 1998.

[43] IhelpYou. "Search engine Partnership Chart ". http ://www.ihelpyou.com/search-engine-chart.html, Février 2005.

[44] Kanatardzic, M. *"Data Mining Concepts, Models, Methods, and Algorithms"*. Wiley - Interscience, 2003.

[45] Kazienko, P., Kiewra, M. "Link Recommendation Method Based on Web Content and Usage Mining", 2003.

[46] Kim, H. "Web Personalization". URL : http ://my.fit.edu/~hokim/file/depthpaper.pdf, Department of Computer Sciences Florida Institute of Technology, Janvier 2005.

[47] Kimball, R., Ross, M. *"Entrepôts de données : Guide pratique de modélisation dimensionnelle (deuxième edition)"*. Vuibert, 2003.

[48] Konopnicki, D., Shmueli, O. "Information gathering in the World-Wide Web : the W3QL query language and the W3QS system". *ACM Trans. Database Syst.*, 23(4) :pages 369–410, 1998.

[49] Kosala, R. Blockeel, H. "Web Mining Research : A Survey". *SIGKDD : SIGKDD Explorations : Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, 2, 2000.

[50] Leavitt, N. "Data Mining for the Corporate Masses?". *Computer*, 35(5) :pages 22–24, 2002.

[51] Lempel, R. , Moran, S. "SALSA : the stochastic approach for link-structure analysis". *ACM Trans. Inf. Syst.*, 19(2) :pages 131–160, 2001.

[52] Li, L., Shang,Y. , Zhang, W. "Improvement of HITS-based algorithms on web documents". In *WWW '02 : Proceedings of the eleventh international conference on World Wide Web*, pages 527–535. ACM Press, 2002.

[53] Maarek, S., Ben Shaul, I. "Automatically organizing bookmarks per content". 1996.

[54] Manley, S. "An analysis of Issues Facing World Wide Web Server". 1997.

[55] Microsoft. "Log file". URL : http ://msdn.microsoft.com/library/, Février 2005.

[56] Mobasher, B., Cooley, R., Srivastava, J. "Automatic personalization based on Web usage mining". *Communication ACM*, 43(8) :pages 142–151, 2000.

[57] Ngu, D., Wu, X. "SiteHelper : A Localized Agent that Helps Incremental Exploration on the World Wide Web". *Proceedings of the 6th International World Wide Web Conference (WWW'97)*, pages 691–700, 1997.

[58] Page, L., Brin, S., Motwani, R., Winograd, T. "The PageRank Citation Ranking : Bringing Order to the Web". Technical report, Stanford Digital Library Technologies Project.

[59] Pal, S., Talwar, V., Mitra, P. "Web mining in soft computing framework : relevance, state of the art and future directions". *Neural Networks, IEEE Transactions on*, 13(5) :pages 1163–1177, 2002.

[60] Pant, S., Bradshaw, G., Menczer, F. "Search engine-crawler symbiosis : Adapting to community interests", 2003.

[61] Pasquier, N., Bastide, Y., Taouil, R. Lakhal, L. "Discovering Frequent Closed Itemsets for Association Rules". In *ICDT '99 : Proceeding of the 7th International Conference on Database Theory*, pages 398–416. Springer-Verlag, 1999.

[62] Pazzani,M., Muramatsu, J., Billsus, D. "Syskill & webert : Identifying interesting web sites". 1996.

[63] Perkowitz, M., Etzioni, O. "Towards adaptive web sites : Conceptual framework and case study". *Computer Networks*, 31.

[64] Porter, M. "An algorithm for suffix stripping". pages pages 130–137, 1980.

[65] Queen's University of Belfast. Parallel computer centre. URL : http://www.pcc.qub.ac.uk/tec/courses/datamining/stu_notes/dm_book_1.html, Janvier 2005.

[66] Risvik, K., Michelsen, R. "Search engines and web dynamics", 2002.

[67] Schechter, S., Krishnan, M., Smith, M. "Using path profiles to predict HTTP requests". *Computer Networks and ISDN Systems*, 30 :pages 457–467, 1998.

[68] SES. "Search Engine Showdown". URL : http://www.searchengineshowdown.com, Janvier 2005.

[69] SEW. "Search Engine Watch". URL : http://www.searchenginewatch.com/, Janvier 2005.

[70] Soft Computing Group. "Data Mining". URL : http://www.softcomputing.com/offres_datamining.html, Janvier 2005.

[71] Souza, J., Matwin, S., Japkowicz, N. "Evaluating Data Mining Models : A Pattern Language". In *Proceedings of the 9th Conference on Pattern Language of Programs (PLOP'2002)*, 2002.

[72] Srivastava, J., Cooley, R., Deshpande, M., Tan, P. "Web Usage Mining : Discovery and Applications of Usage Patterns from Web Data". *SIGKDD Explorations*, 1(2) :pages 12–23, 2000.

[73] Sung, H. "Helping Online Customers Decide through Web Personalization". *"Proceedings of the WEBKDD Workshop : Web Mining for Usage Patterns and User Profiles, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining"*, 17(6) :pages 34–43, Décembre 2002.

[74] Surfaid. Surfaid analytics. URL : http ://surfaid.dfw.ibm.com/, Février 2005.

[75] Wang, Y. "Web Mining and knowledge Discovery of usage patterns". URL : http ://db.uwaterloo.ca/~tozsu/courses/cs748t/surveys/wang.pdf, Janvier 2005.

[76] Wen, J. , Nie, J., Zhang, H. "Query clustering using user logs". *ACM Transactions on Information Systems (TOIS)*, 20(1) :pages 59–81, 2002.

[77] Wen, J., Nie, J., Zhang, H. "Clustering user queries of a search engine". *Proceedings of the 10th International World Wide Web Conference (WWW'01)*, pages 162–168, 2001.

[78] Whalen, D. "The Unofficial Cookie FAQ version 2.6". URL : http ://www.cookiecentral.com/faq/, Février 2005.

[79] Wu, K. , Yu, P. , Ballman, A. "Speedtracer : A web usage mining and analysis tool". *IBM Systems Journal*, 37(1), 1998.

[80] Yianilos, P. "The LikeIt intelligent string comparison facility". 1997.

[81] Zaiane, O., Han, J. "WebML : Querying the World-Wide Web for Resources and Knowledge". In *Workshop on Web Information and Data Management*, 1998.

[82] Zaine, O., Xin, M., Han, J. "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs". In *Advances in Digital Libraries*, pages 19–29, 1998.