Système d'aide à la reformulation de requêtes réparties sur les sources d'information hétérogènes

Liang DONG*,Bernard DOUSSET*, Christel PORTE**,Christian LONGVIALLE**
{dong/dousset}@irit.fr , {porte/longevia}@univ-mlv.fr

* Institut de Recherche en Informatique de Toulouse, Equipe SIG, Université Paul Sabatier
118, route de Narbonne 31062 Toulouse cedex 4
** ISIS/CESD Université de Marne la Vallée, avenue Albert Einstein, champs sur marne 77000

Mots clefs :

Systèmes de Recherche d'Information, Reformulation de requête, Tétralogie, AFC

Keywords :

Information Retrieval System (IRS), Query reformulation, Tetralogie, AFC

Palabras claves :

Sistema de a recupercion de datos, reformulacion de la pregunta, Tetralogie, AFC

Résumé :

A la suite de l'évolution rapide des réseaux de communication, de l'avènement d'Internet et de la mise en ligne de bases de données sur la toile, un gigantesque espace d'information hétérogène est devenu accessible à tous et son volume augmente considérablement chaque année. Actuellement, nous pouvons accéder plus ou moins efficacement à l'information via les bases de données, les systèmes de recherche d'informations ou les systèmes de découverte de connaissances. Face à ce flot d'informations, des outils d'interrogation spécifiques, le plus souvent booléens, permettent théoriquement aux utilisateurs de rechercher et de trouver l'information souhaitée. Cependant, pour l'utilisateur, la tâche d'interrogation reste particulièrement difficile car celui-ci n'a pas toujours recours aux termes nécessaires qui lui permettraient d'exprimer ses besoins en information de façon pertinente. Les réponses s'avèrent donc peu satisfaisantes, car trop nombreuses, bruitées (restitution de documents non pertinents), peu précises et bien souvent incomplètes. Pour améliorer les performances de la recherche, nous proposons un système d'interrogation qui doit permettre une utilisation plus aisée et surtout plus efficace des différentes sources d'informations actuellement disponibles (bases locales, CD/ROM, serveurs classiques et Internet : web, news, wais). Ce système est basé sur la reformulation de requête et s'appuie sur une étude de concordance entre termes dans un ensemble de documents de référence.

Abstract :

With the fast development of the networks and database online, a gigantic space of heterogeneous information become available and its volume increases considerably each year. At present, we can effectively retrieve information through databases, Information Retrieval System and knowledge discovered systems. To overcome the problem of flood information, we need specific tools, generally Boolean, that allow users to get their information efficiently and quickly. However, as a user, the task of interrogation remain particularly difficult because it is not always very easy for him to find the correct terms to express his requires, therefore sometimes the answers, which are impure or incomplete are not very satisfying. To improve the performances of retrieve, we propose a system of interrogation which allows users to use various available information sources easily and effectively. This system is based on the query reformulation and it uses a study of the concordance between terms of reference documents group.

1. Introduction

Depuis déjà plusieurs années, le volume des données électroniques croît d'une façon exponentielle, aussi bien sur Internet que pour les besoins stratégiques ou internes des entreprises. Aujourd'hui Google affiche 3 milliards de pages. Les données utiles à l'entreprise se présentent sous des formes très diverses et pas nécessairement compatibles entre elles. Par exemple, on accède aux informations scientifiques depuis les bases documentaires (Inspec, Biosis, Medline,...) disponibles en ligne sur les grands serveurs (Dialog, Questel,...) ou sur CD-ROM, aux données technologiques par les diverses sources de brevets proposées, tandis que, sur Internet, on peut aussi rechercher les informations économiques ou financière dans la presse électronique spécialisée (le Monde économique, les Echos,…). Dans ce contexte, la majorité des Systèmes de Recherche d'Information (SRI) présente, en réponse à la requête de l'utilisateur, des résultats sous forme d'une liste linéaire de documents parfois ordonnée en fonction de leur pertinence relative. Mais, le plus souvent, la qualité de l'extraction n'est pas au niveau escompté. Or, la recherche de documents dans une base de données est un processus clé dans l'exploitation des ressources en information, elle est une activité quotidienne très largement pratiquée par divers types d'utilisateurs, c'est une tâche assez complexe qui nécessite la mise en relation d'un besoin d'information imprécis avec le contenu d'une multitude de documents souvent pas très bien indexés. Elle dépend étroitement de la capacité de l'utilisateur à définir son besoin d'information et à élaborer une stratégie de recherche pour affiner et/ou élargir les résultats obtenus. Mais, les utilisateurs ne sont que très rarement des professionnels de la documentation et ils ne trouvent pas toujours les termes nécessaires qui leur permettraient d'exprimer pleinement, dans leur *équation de recherche*, leurs besoins spécifiques en terme d'information.

1.1 Principaux problèmes de la recherche d'information

- trop de résultats [1]

Il arrive fréquemment que la liste des documents restituée par les moteurs de recherche ou les systèmes d'interrogation soit trop large, bruitée (restitution de documents non pertinents) et peu précise : faible taux de précision.

La difficulté est ici double: d'une part un mot a plusieurs sens possibles, sens qui ne se précise que par l'adjonction d'un autre mot qui en fait une expression (syntagme), ou par l'addition d'un contexte d'utilisation (terme cooccurent). D'autre part, au sein d'une base documentaire, un concept donné n'est généralement pas exprimé de manière unique et homogène (non-uniquely identified content).

C'est simplement parce qu'il existe plusieurs manières de dire la même chose et parce que pour des questions de style ou de précision de son propos, un rédacteur s'efforce généralement de varier ses formulations, en utilisant des expressions synonymes, hyperonymes ou hyponymes.

Cet inconvénient majeur est dû au fait que la majorité de l'information présentée dans une source est typiquement non-structurée, c'est à dire non-classée et non-réferencée par les moteurs autrement que mot par mot. Or, un mot isolé est par essence ambigu, et seul son contexte d'utilisation permet de préciser son sens exact. Il en résulte que la plupart du temps, et quelque soit le volume initial de la source, l'utilisateur doit nécessairement consulter un nombre important de documents pour trouver ce qu'il cherche.

Outre cela, à cause du même article avec le même contenu peut être présenté sous des formes différentes comme html, pdf, post-script, il y a de plus en plus de résultats retournés.

- peu de résultats

Le pire est que, très souvent, l'utilisateur saisit une requête trop précise et risque de ne pas obtenir un ensemble de documents pertinents pour sa requête (effet de silence) : d'où un faible rappel. C'est surtout pour résoudre ce problème, que nous voulons reformuler la requête initiale. Car dans notre cas, le but avoué est une certaine exhaustivité afin de couvrir au mieux un domaine scientifique ou technique.

- les résultats ne sont pas tous pertinents et l'information retrouvée n'est pas complète

Le ou les mots clés saisis par l'utilisateur ne correspondent que rarement à la formulation efficace et en accord avec le contenu des documents recherchés. Cela conduit généralement à la mise en avant de résultats non pertinents car trop éloignés du sujet de la recherche, ainsi qu'à la disparition de résultats pourtant pertinents et effectivement disponibles sur la source. Les documents recherchés sont alors noyés dans une part non négligeable de bruit.

1.2 Modèles de recherche

Dans ce contexte, de nombreux de travaux ont été proposés:

- La stratégie d'expansion de requête [2], [3], son principe fondamental est de comparer simplement le contenu de la requête avec les documents de la base. L'ensemble des documents pertinents restitué est alors très souvent incomplet. Des travaux de recherche ont proposé, d'ajouter d'autres termes contenus dans les documents pertinents ou d'ajouter des termes sémantiquement proches ou encore d'ajouter des terme voisins en utilisant des calculs de poids de similarité entre termes.

- La reformulation par retour de pertinence. Son principe fondamental est de formuler la requête initiale pour amorcer la recherche d'information, puis itérativement la modifier à partir des jugements de pertinence et/ou de non-pertinence de l'utilisateur afin d'ajuster la requête par expansion, repondération ou combinaison des deux procédures, jusqu'à ce que le résultat de la recherche soit satisfaisant. Dans ce cadre, de nombreux modèles de recherche ont été proposés : le modèle vectoriel [4], [5] ; le modèle probabiliste [6], [7], [8] ; le modèle connexionniste [9], [10], [11].

- La technique des algorithmes génétiques [12], [13], [14] est largement adaptée à la reformulation de requête. Ces algorithmes sont inspirés des mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des individus les mieux adaptés et ceux de la propagation du patrimoine génétique. Leur but est de faire évoluer un ensemble d'individus candidats à un problème posé vers un individu optimal.

Le projet décrit ici vise à offrir un outil d'aide à la reformulation de requêtes. Notre outil repose sur des techniques d'analyse de données, d'analyse de l'évolution et de recherche de corrélations. Il permet à l'utilisateur d'analyser les résultats, de sélectionner les termes significatifs, d'affiner sa demande, de retrouver les résultats sous forme de nouvelles connaissances.

Dans le second chapitre de cet article, nous présentons l'étude de l'existant qui donne un aperçu des techniques de reformulation de requête. Le troisième chapitre détaille la description de notre méthode de reformulation de requête. Enfin, nous présentons l'implantation de cette méthode ainsi que certains résultats issus d'expérimentations.

2. Principe de la reformulation de requête

La reformulation de requête est proposée comme un processus qui a pour objectif de générer une nouvelle requête plus adéquate afin d'obtenir un ensemble de résultats plus pertinent, à partir de connaissances du domaine cible, en utilisant les concepts clés contenus dans les documents. La requête initiale est formulée par l'utilisateur, nous l'utilisons pour amorcer la recherche, puis nous la modifions par ajout de termes significatifs, suppression de termes non pertinents et/ou en réestimant leurs poids. Cette modification s'effectue de manière itérative après exploration des résultats des requêtes intermédiaires. Le "retour de pertinence" ("relevance feedback", en anglais) est le nom donné à la méthode de modification automatique de la requête permettant cette fonctionnalité. La nouvelle requête obtenue à chaque itération de "feedback", permet de réorienter la recherche vers les documents pertinents.

Le processus traditionnel de la reformulation de requête est souvent présenté comme un cycle d'activités. La figure 1 suivante montre ce processus de recherche d'information.

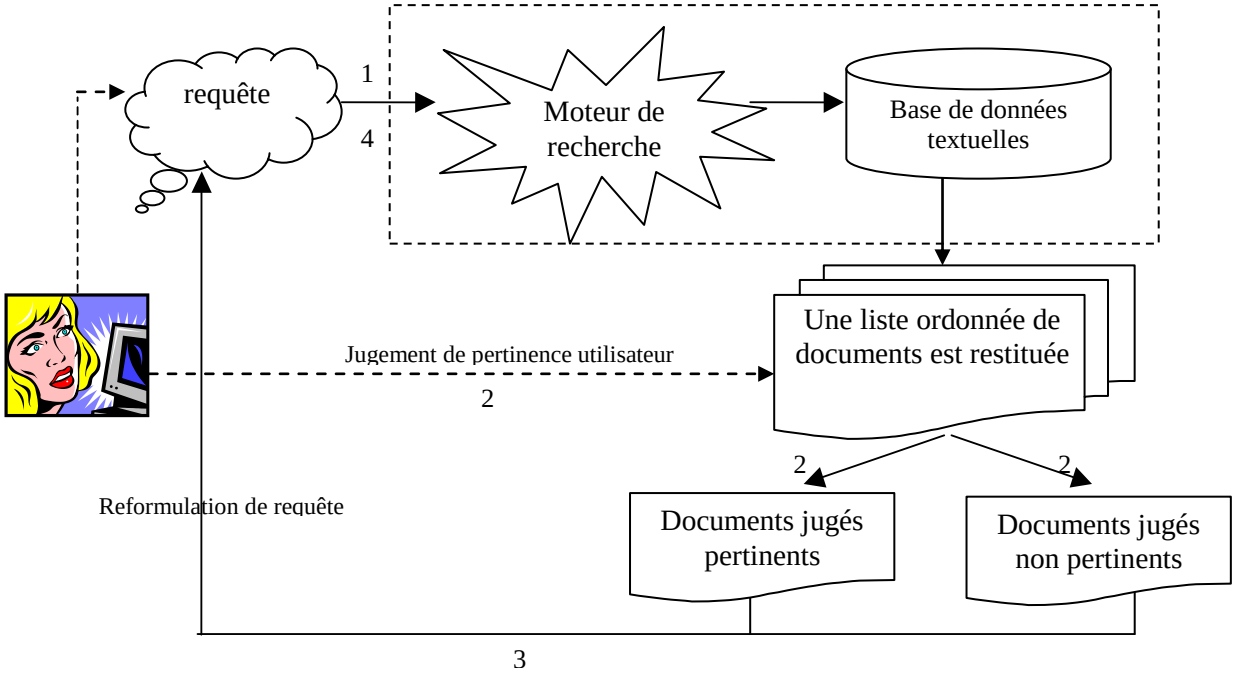

Figure 1. Processus de Recherche d'information

Le processus présenté est constitué de 4 étapes principales:
- **Etape 1** : L'utilisateur propose une requête sur un moteur de recherche ou un système de recherche d'information. Le système de recherche présente, en retour à l'utilisateur, un ensemble de documents.

- **Etape 2** : L'utilisateur peut sélectionner quelques documents qu'il considère comme pertinents.

- **Etape 3** : Le système reformule automatiquement la requête, en fonction du jugement de pertinence de l'utilisateur sur les documents déjà proposés. Enfin, il produit une nouvelle requête. Cette nouvelle requête associe de nouveaux termes à la requête initiale. Ces nouveaux termes sont extraits des documents retrouvés (après une première recherche) et les opérateurs mis en œuvre sont choisis en fonction de la distribution des termes dans les documents.

- **Etape 4** : Relancer cette requête modifiée, le système présente à l'utilisateur un nouvel ensemble de documents retrouvés. On retourne à l'étape 2 en direction des documents pertinents.

Actuellement la technologie la plus répandue est l'interrogation par requête dite booléenne associée à la visualisation des résultats dans une liste ordonnée (selon différents critères comme la pertinence, la date de création,…).

De nombreux travaux se sont intéressés à l'intégration des approches de retour de pertinence dans les différents modèles de recherche. Par exemple, le serveur AltaVista (www.ac-montpellier.fr) mettait à la disposition de ces utilisateurs un outil de reformulation de requêtes nommé la fonction Refine (anciennement appelée Live Topics). La fonction Refine est un puissant outil, propre à AltaVista, qui prend en compte l'ensemble des pages résultant d'une requête d'un utilisateur, puis analyse leur contenu et propose des mots-clés en rapport avec l'interrogation de départ, c'est-à-dire les mots revenant le plus souvent dans les pages - résultats. Ces mots-clés potentiels sont classés par thèmes, puis, à l'intérieur de chaque thème, par ordre de fréquence décroissante des mots dans les pages analysées. L'utilisateur peut alors affiner sa requête en choisissant ou en éliminant certain des mots-clés, grâce à deux possibilités : Require et Exclude. C'est-à-dire, ces nouveaux mots clés vous sont proposés et vous pouvez les exclure ou les inclure dans la suite de votre recherche ; les autres mots, pour lesquels vous n'aurez choisi ni exclusion, ni inclusion seront ignorés. Une fois cette manipulation effectuée, lancez la nouvelle requête ainsi définie !
UMAP, qui est un logiciel de traitement et de visualisation d'informations volumineuses très connu, analyse des données en provenance d'Internet et de sources internes à l'entreprise et en extrait l'ensemble des mots les plus fréquemment utilisés. Il représente alors, sous forme de carte, les

connaissances contenues dans le corpus analysé. L'utilisateur « navigue » sur cette carte selon ses centres d'intérêt et peut ainsi s'approprier plus facilement l'information qui lui est utile.
Les applications de ces produits sont populaires et mûres. Néanmoins, de plus en plus de techniques innovantes apparaissent dans ce domaine.

3. Présentation de notre méthode

Notre nouvelle technologie doit permettre à des non-spécialistes d'utiliser une interface visuelle pour retrouver des informations pertinentes dans de grandes bases de données textuelles, que celles–ci soient le Web lui-même ou bien des bases de données disponibles sur l'Intranet.
La figure 2 présente le contexte et l'enchaînement du processus.

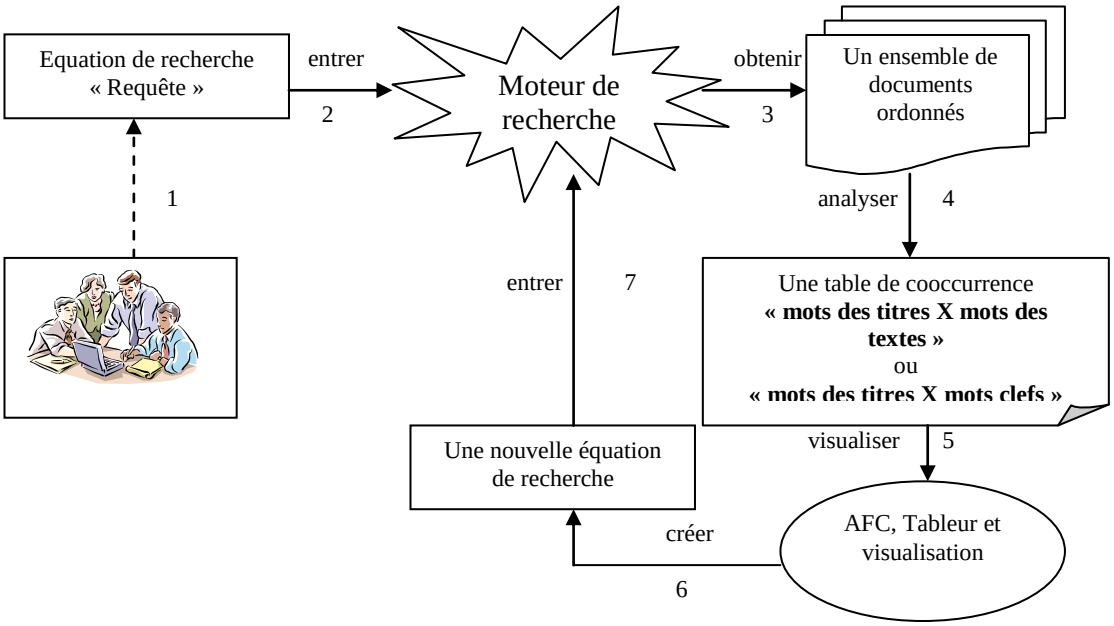

Figure 2. Le contexte de notre modèle d'exploitation

1. l'utilisateur détermine une équation de recherche (requête). La définition du mot « requête » dans le Dictionnaire Francophone de l'Informatique est exprimée comme suit :
Une requête : Question posée à une base de données concernant les informations qu'elle contient.
Dans le système de recherche d'information, elle correspond à l'expression du besoin en informations de l'utilisateur, dans le formalisme de représentation de connaissances.
Le processus de transformation de la requête dans le modèle booléen donne la possibilité aux utilisateurs de formuler/reformuler naturellement leurs requêtes et d'autre part facilite la consultation des tables d'index et la sélection des documents solutions car le modèle booléen s'adapte bien aux langages scientifiques ou techniques qui ont l'avantage d'être très descriptifs.
2. l'utilisateur lance la requête dans un moteur de recherche ou une base de donnée scientifique.
3. l'utilisateur obtient en retour un ensemble de documents. Cependant, nous choisissons quelques documents pertinents qui sont jugés par l'utilisateur - que nous appelons « DOCUMENTS NATIFS » concernant le sujet à explorer.
Je ne présente pas les points 1 à 3 de cette méthode. Car ceci dépend du moteur de recherche et le jugement d'utilisateur, mais je présente de façon détaillée les autres points.
4. à partir des documents « natifs », nous éliminons automatiquement certaines données inutiles (ex : des mots « vides »), créons de dictionnaires de synonymes et sur le « seuillage » qui efface le contenu des cellules dont la valeur est en dessous d'un seuil donné (déconnexion partielle du graphe des liaisons). Ensuite, il est automatiquement crée une matrice de cooccurrence **mots des titres X mots des textes** ou **mots des titres X mots-clefs**. Nous illustrons ceci par la figure suivante :

Figure 3. Mise en évidence d'un cluster dans le tableau

5. Enfin, nous procédons :

- soit à une Analyse Factorielle des Correspondances (AFC) qui permet d'étudier les liens existants entre les termes correspondants aux lignes et aux colonnes de la matrice choisie,
- soit à une classification par blocs de cette même matrice si le nombre de colonnes est trop grand pour appliquer de façon efficace la première méthode.

Le but de l'AFC est de donner une (ou plusieurs) représentation(s) carte(s) des fréquences relatives d'un tableau de contingence selon les lignes et selon les colonnes, puis d'établir des correspondances ou liaisons entre les représentations des lignes et les représentations des colonnes. Ici, nous utilisons ces cartes pour visualiser le nuage « sémantique » central et les nuages périphériques.
Un exemple de carte est donné ci-après :

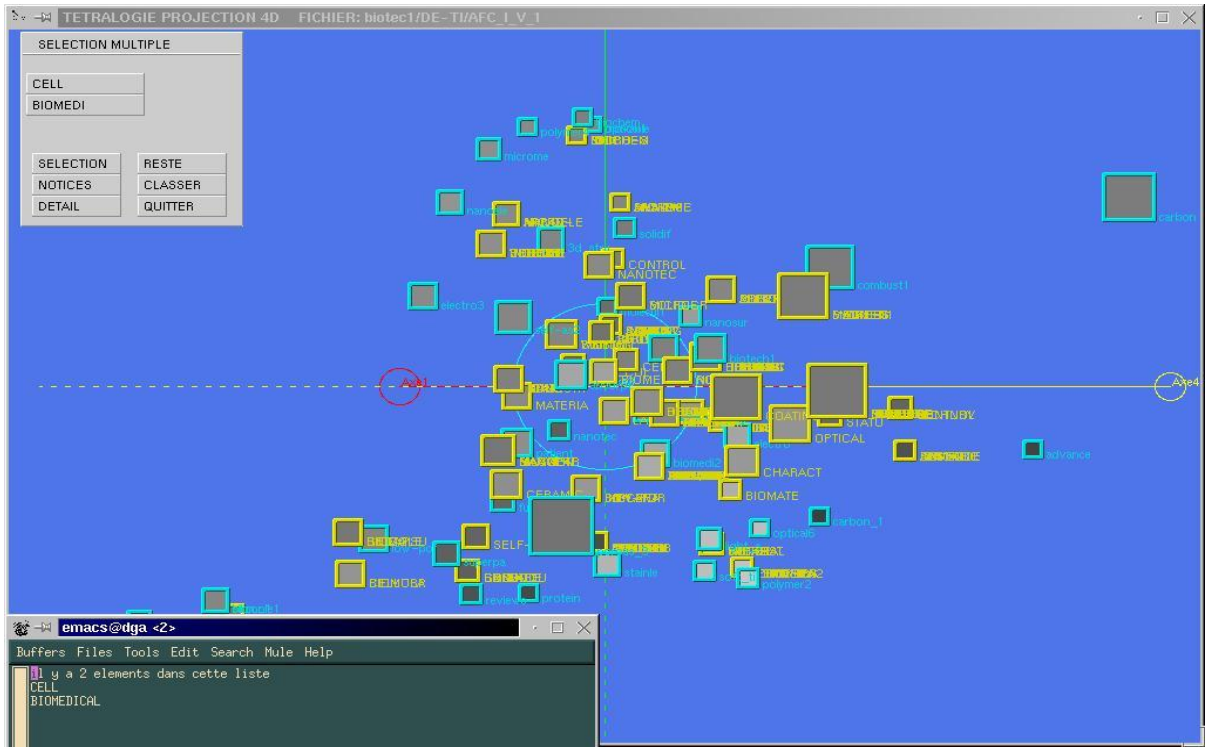

Figure 4. Carte factorielle de l'AFC

6. nous établissons l'équation de recherche. Pour établir l'équation, qui doit permettre la collecte d'informations semblables aux documents natifs, il suffit de traduire la lecture visuelle des cartes ou des classes obtenues en une suite d'opérateurs logiques.

La lecture s'effectuera du centre des cartes vers l'extérieur ou du haut de la diagonale vers le bas.

a) équation liée au nuage central :
Les points (individus et variables) constituant le nuage central sont les mots génériques du thème recherché. Nous pouvons trouver ces mots dans la carte de l'AFC et afficher ces mots. Si on continue comme l'exemple précédent, l'application de cette méthode donne une liste de mots, comme la figure suivante :

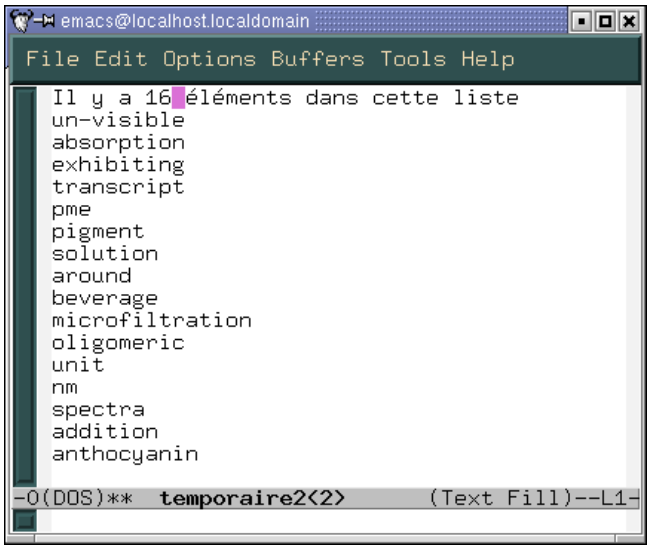

Figure 5. Mise en évidence d'une liste de mots par rapport une classe donnée

Désignons par m_{ci}^{0}, avec i ∈ {1, 2, …, n}, les mots des colonnes constituant le nuage de points situés à l'origine.
Désignons par m_{lj}^{0}, avec j ∈ {1, 2, …, p}, les mots des lignes constituant le nuage de points situés à l'origine.

i) on associe les variables m_{ci}^{0} entre-elles par la fonction logique « OU ».
ii) par la fonction logique « ET », on associe à l'entité créée en i) à l'ensemble des individus m_{lj}^{0} liés entre eux par des fonctions logiques « OU ».

On obtient ainsi l'expression de l'équation de recherche liée au nuage central que l'on note ❶ :

$$❶ \equiv (m_{c1}^{0} \text{ OU } m_{c2}^{0} \text{ OU } \ldots \text{ OU } m_{ci}^{0} \text{ OU } \ldots \text{ OU } m_{cn}^{0}) \text{ ET } (m_{l1}^{0} \text{ OU } m_{l2}^{0} \text{ OU } \ldots \text{ OU } m_{lj}^{0} \text{ OU } \ldots \text{ OU } m_{lp}^{0})$$

<u>Remarque :</u>

Pour les noms composés, encore appelés multi-termes (par exemple : composants électroniques), il est utilisé les opérateurs de proximité AV (pour « Avec » en français) pour les serveurs français et W (pour « With » en anglais) sur les serveurs anglophones.

b) équations liées aux nuages périphériques :
Il est procédé à une exploration systématique des nuages de points extérieurs au centre. Nous offrons la possibilité d'observer l'espace tout entier en visualisant les groupes les plus forts présents sur les quatre premiers axes et en glissant progressivement vers des groupes contenus dans les autres sous-espaces.
Chaque nuage ainsi découvert est analysé par la connaissance des mots constituant ses colonnes et ses lignes et donne lieu, par application de la méthode décrite précédemment, à l'élaboration d'une équation.
Ainsi pour le premier nuage découvert, on aura l'équation notée ❷ :

$$❷ \equiv (m_{c1}^{1} \text{ OU } m_{cé}^{1} \text{ OU } \ldots \text{ OU } m_{ci}^{1} \text{ OU } \ldots \text{OU } m_{cq}^{1}) \text{ ET } (m_{l1}^{1} \text{ OU } m_{l2}^{1} \text{ OU } \ldots \text{ OU } m_{lj}^{1} \text{ OU } \ldots \text{ OU } m_{lr}^{1})$$

Pour le second nuage découvert, on aura l'équation ❸ :

❸ ≡ (m^2_{c1} OU $m^2_{cé}$ OU … OU m^2_{ci} OU … OU m^2_{cs}) ET (m^2_{l1} OU $m^é_{l2}$ OU … OU m^2_{lj} OU …OU m^2_{lt})

Et ainsi de suite jusqu'à épuisement de la visualisation.

c) équation finale de recherche:
La requête finale servant à l'interrogation en langage booléen des différentes sources d'informations sera une combinaison logique des différentes équations élaborées précédemment, à savoir :

❶ ET (❷ OU ❸ OU ❹ OU ❺ OU ……)

7. l'utilisateur relance cette nouvelle requête, le processus continue jusqu'à la satisfaction des besoins d'information.

4. Validation de notre méthode

Nous présentons dans cette section les différents incréments que nous avons réalisés, dans le but de valider notre méthode. Nous évaluerons de l'aide à la reformulation des requêtes, elle sera focalisée sur la « qualité » des termes extraits et de la pertinence des documents obtenus déterminera la qualité de l'aide offerte pour la vérification des résultats des recherches. La pertinence des termes est quantifiée en fonction de son utilité.

4.1. L'environnement

Les tests ont été effectués sur une base de données scientifique qui était disponible dans notre équipe. Cette base contient plus de 4000 documents couvrant le domaine des maladies nosocomiales.
Le projet est développé suivant le cycle incrémental suivant :

4.2. Incrément 1

4.2.1. Les caractéristiques de la première incrémentation

Dans les documents pertinents, nous supprimons des mots « vides » et nous créons une matrice de cooccurrence « mots du texte X mots du texte » ;
Nous filtrons les éléments de la matrice $a_{i,j}$, lorsque $a_{i,j}$ <=1 et produisons automatique une nouvelle matrice;
A partir de cette matrice, nous faisons une matrice de proximité en fonction de la formule

$$a_{i,j} = \frac{a_{i,j}}{\sqrt{a_{i,i}} \times \sqrt{a_{j,j}}} \ ;$$

Nous trions ensuite cette matrice de proximité par blocs en mode relatif et construisons des classes ;
Dans chaque classe, nous extrayons des mots pour composer les équations de recherche ;
Enfin, nous combinons toutes les équations de recherche et relançons l'équation finale.

4.2.2. Les résultats préliminaires

Dans cette expérimentation préliminaire, nous avons sélectionné uniquement les dix premiers documents restitués par la base de données. Cette expérimentation une fois réalisée permet d'obtenir les principaux résultats suivants :

1. une matrice de cooccurrence

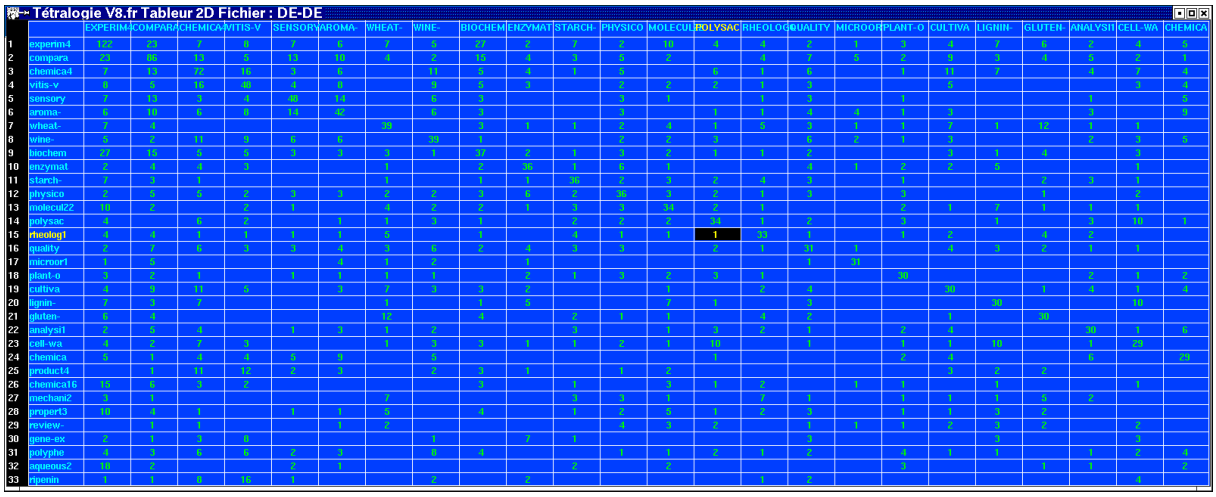

Figure 6. Matrice de cooccurrence « mots du texte X mots du texte »

2. lorsque cette matrice a été triée, nous pouvons, comme ci-dessous, visualiser sa structure par un zoom

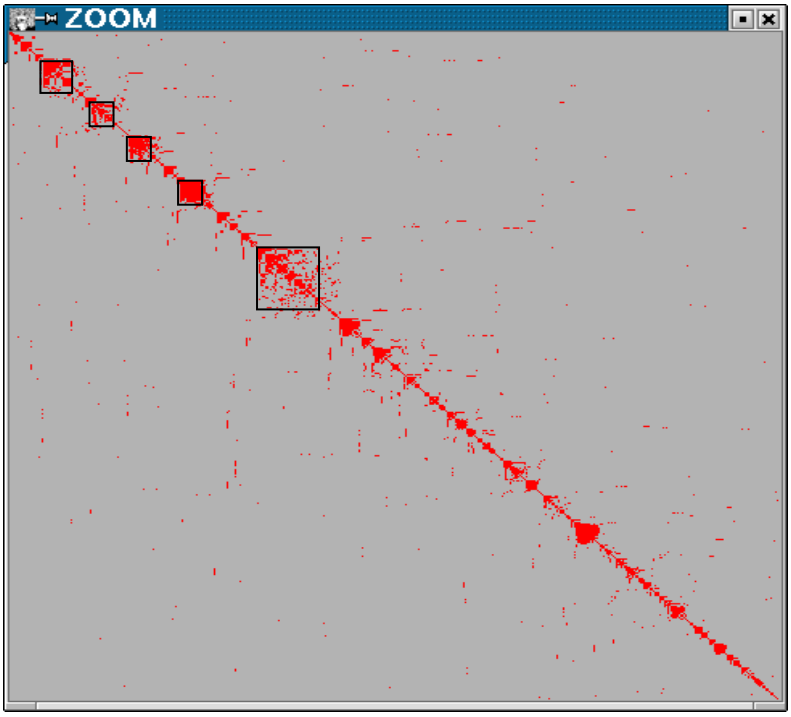

Figure 7. Mise en évidence d'un zoom de la matrice

3. à partir cette représentation graphique, nous pouvons trouver des blocs (classes). Chaque sommet présent séparément une classe. Au sein d'une classe, les mots sont reliés entre eux par des liens plus ou moins forts calculés en fonction des cooccurrences relatives des mots dans les textes. Ci-dessous, nous illustrons une figure, nous donnons un exemple de classe.

Figure 8. Matrice de proximité par blocs en mode relatif

4. Au sein d'une classe, nous pouvons récupérer des mots importants. Ces mots représentent le contenu du corpus textuel. La sélection d'un mot permet de retourner aux documents qui lui sont liés. Nous utilisons ces mots pour établir l'équation d'intégration afin d'aider à la reformulation d'une requête documentaire.

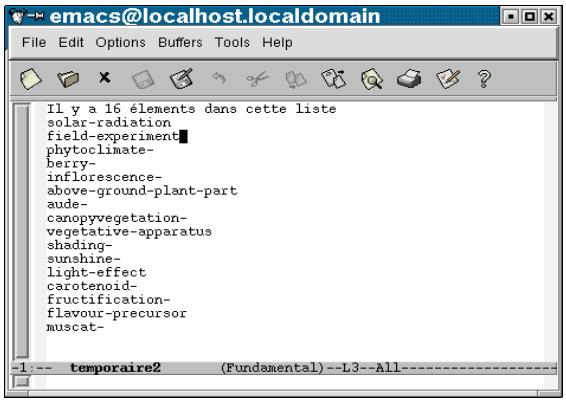

Figure 9. Mise en évidence d'une liste de mots par rapport une classe donnée

Dans la figure suivante, nous montrons des étapes pour extraire des mots.

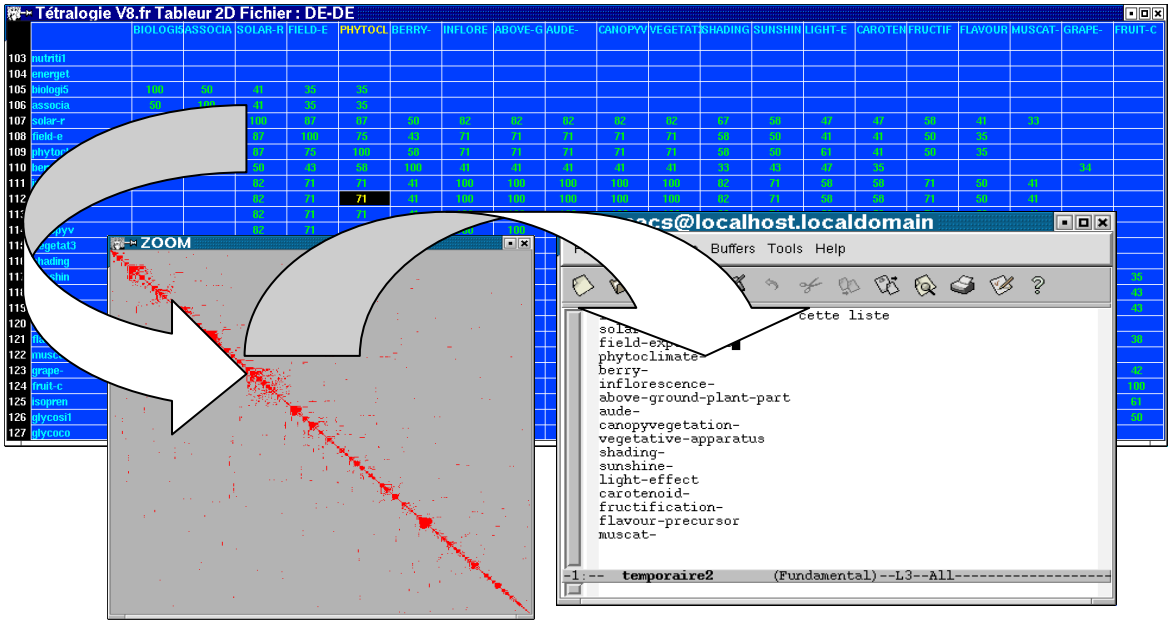

Figure 10. Les étapes pour extraire des mots

4.3. Incrément 2

4.3.1. L'objectif du second incrément

Le second incrément développe la stratégie complémentaire d'élaboration de la requête qui ne fait plus maintenant appel aux méthodes classiques utilisées par les documentalistes pour interroger les banques de données scientifiques ou techniques.

4.3.2. Les résultats préliminaires

L'étape de validation de notre méthode est basée sur plus d'une dizaine de documents venant d'une base de données du domaine des maladies nosocomiales. Chaque document contient des informations de type bibliographique comme par exemple : auteurs, titre, résumé, descripteurs, etc.
Nous générons, tout d'abord, une matrice équivalente à celle de la figure 6 en filtrant les mots vides et en utilisant, si nécessaire, un dictionnaire de synonymes. Un « seuillage » permet d'éliminer le contenu des cellules dont la valeur est jugée trop faible pour représenter une concordance valide. Plusieurs matrices, croisant deux champs sémantiques, peuvent être utilisées dans notre démarche : « Mots-clés X Titres », « Résumés X Mots-clés », etc, mais la plus efficace, pour l'instant, reste « Résumés X Titres ».
A partir d'une de ces matrices initialement triées par consistance, un tri par blocs diagonaux effectué en mode absolu permet de dégager un premier cluster, qui représente habituellement le sujet central du domaine étudié. La figure suivante met bien en évidence ce type d'agrégation, qui apparaît dans le coin « nord-ouest » de la matrice.
Les mots correspondants aux colonnes du cluster représentent les m^0_{ci}, avec i $\in$ {1, 2, …, n} et les mots correspondants aux lignes les m^0_{lj}, avec j $\in$ {1, 2, …, p}. Après élimination des doublons entre lignes et colonnes, nous obtenons alors l'expression de l'équation de recherche liée au nuage central :

❶ $\equiv (m^0_{c1}$ OU m^0_{c2} OU … OU m^0_{ci} OU … OU $m^0_{cn}) $ ET $(m^0_{l1}$ OU m^0_{l2} OU … OU m^0_{lj} OU … OU $m^0_{lp})$

Tétralogie V7.fr Tableur 2D Fichier : abs-abs

		hospital	infections	adults	brazilian	comparing	elderly	university	younger	brazil	clinicas	isolated	nosocomial	antimicrob
1	infections	18	11	8	8	8	8	8	8	2	2	5	3	2
2	patients	16	11	8	8	8	8	8	8			3	4	1
3	nosocomial	14	10	7	7	7	7	7	7			3	5	1
4	hospital	14	5	5	5	5	5	5	5	4	4	4		
5	infection	9	5	4	4	4	4	4	4	1	1	2	2	3
6	antibiotics	6	4	3	3	3	3	3	3			1	1	1
7	surgical	6	3	3	3	3	3	3	3				1	
8	risk	4	3	2	2	2	2	2	2			1	1	
9	brazil	3	1	1	1	1	1	1	1	1	1	1		
10	genes	2								2	2	2		
11	strains	2	1							2	2	3	1	
12	resistant		5									5	6	
13	antibiotic		3									3	3	6
14	resistance	1	2							1	1	3	2	7
15	pathogens													3
16	care													3
17	intensive													3

Figure 11. Matrice triée par blocs en mode absolue

En suite, nous éliminons ces termes centraux et nous classons le reste des mots en plusieurs clusters (nuages périphériques) grâce, soit à une analyse des correspondances, soit à une analyse des « liaisons relatives» entre les représentants des lignes (par exemples : mots des résumés) et les représentants des colonnes (par exemple : mots des titres).
Après élimination des doublons entre lignes et colonnes, on obtiendra, pour chaque classe, une équation notée ❷ :

❷ $\equiv (m^1_{c1}$ OU $m^1_{cé}$ OU … OU m^1_{ci} OU …OU $m^1_{cq})$ ET $(m^1_{l1}$ OU m^1_{l2} OU … OU m^1_{lj} OU … OU $m^1_{lr})$

A l'issue de ces deux étapes, nous combinons l'équation ❶ et les équations de type ❷ afin de générer l'équation de recherche finale.

❶ ET (❷ OU ❸ OU ❹ OU ❺ OU ……)

L'équation est alors automatiquement réécrite en langage booléen sous la forme suivante :

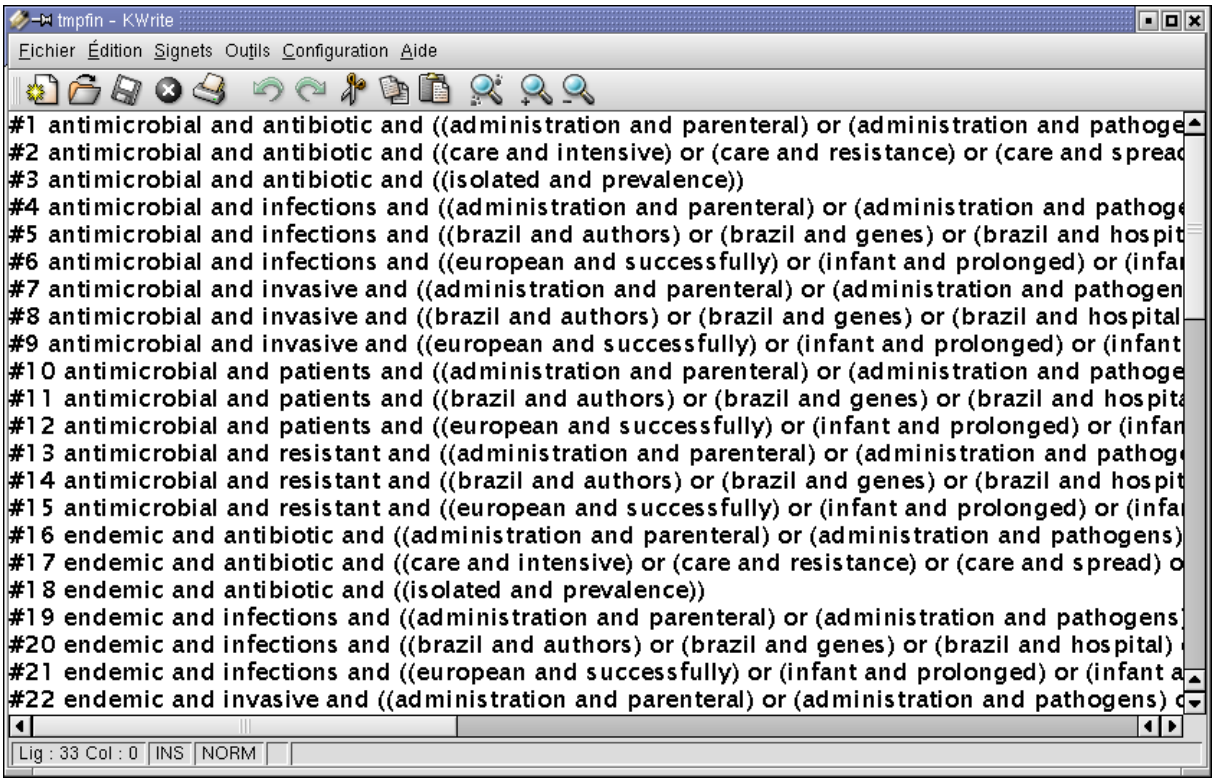

Figure 12. une liste d'équations

Nous devons relancer cette nouvelle équation sur le système de cherche d'information initialement utilisé (par exemple Dialog) ou sur un moteur de recherche (par exemple : Google) ou sur tout autre système utilisant des opérateurs booléens. Mais des limitations dans la taille de l'équation nous obligent souvent à la segmenter (afin, le plus souvent, de ne pas dépasser 256 caractères). Pour l'interrogation des bases bibliographiques (en ligne ou sur CD/Rom) chaque partie de l'équation donne naissance à un résultat intermédiaire nommé *#i*, il est possible d'agréger ensuite l'ensemble de ces résultats par une série de OU, comme ci-dessous :

#1 OU #2 OU #3 OU #4 OU #5 OU ……

Cette nouvelle équation pouvant éventuellement dépasser elle aussi 256 caractères, elle peut à son tour être segmentée, c'est le cas dans l'exemple précédent (Figure 12) qui compte 60 lignes.

Pour les moteurs de recherche, la stratégie est différente car les *#i* n'opèrent plus, les segments d'équation sont alors successivement envoyés au moteur, qui retourne à chaque fois une liste d'URL. Les URL répondues sont souvent les mêmes, elles sont alors classées par fréquences décroissantes, plus haute est la fréquence plus le document correspondant à l'URL semble pertinent. La figure suivante présente une liste d'URL ainsi triée, les plus pertinentes sont en haut.

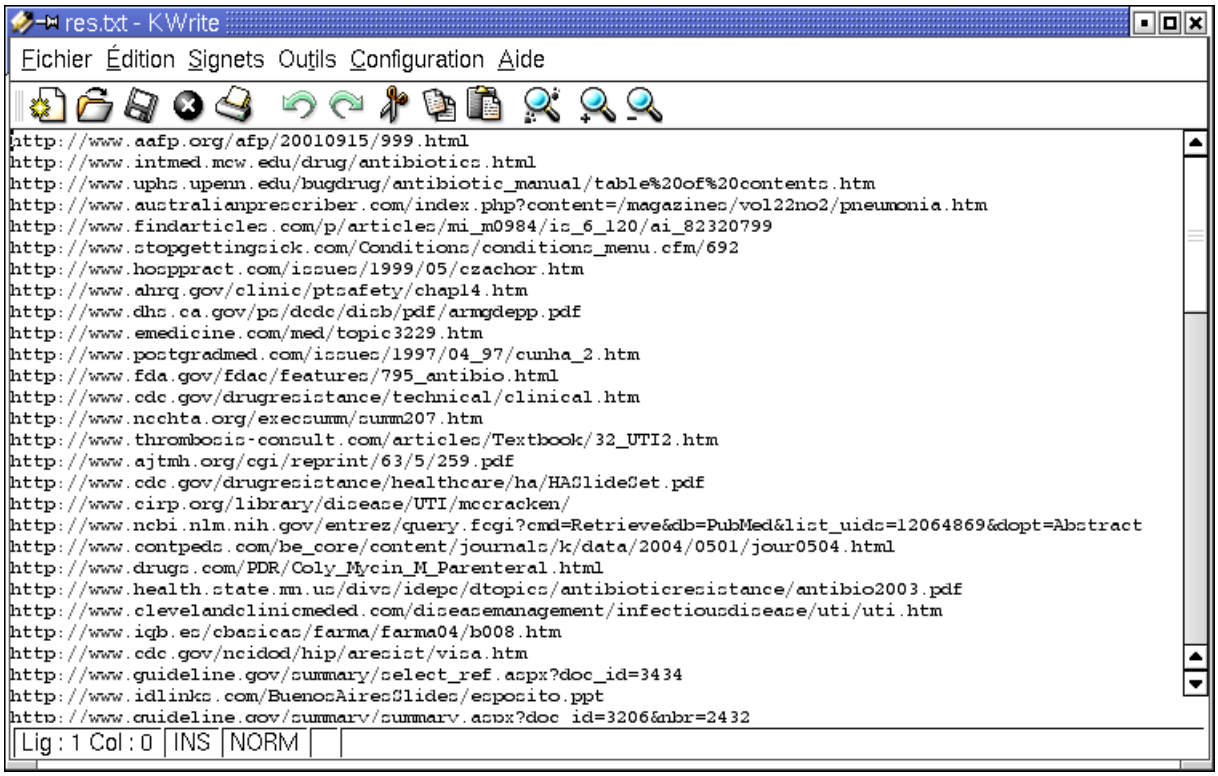

Figure 13. liste d'URL classée par pertinence.

Une fois les documents récupérés, l'utilisateur peut les consulter et juger de leur pertinence. Selon le jugement qu'il porte, nous construisons un nouvel ensemble de « documents natifs » et le processus peut être réitéré jusqu'à la satisfaction des besoins en information.

4.4. Incrément 2 bis

Pour les documents au format Html, l'incrément 2 bis représente une alternative à l'incrément précédent. Nous pouvons décomposer un document Html afin d'en différencier les informations utiles, par exemple : le titre, les mots-clés et les mots du texte… Excepté la forme du document, toutes les étapes sont identiques à la deuxième incrémentation.
Pour réaliser notre expérimentation au format Html, nous avons utilisé la collection de documents du système Mercure. Le système Mercure sert habituellement à récupérer les documents en utilisant les requêtes existantes. En effet, il existe 50 requêtes qui couvrant des domaines variés. La taille de la collection est 2,73 GigaOctets, ce qui correspond à 5164 documents Web.

5. Conclusion

Face à des résultats insatisfaisants et souvent inadaptés, les techniques de Recherche d'Information consistent à trouver, extraire, analyser et sélectionner une collection de documents pertinents pour un besoin exprimé par le demandeur d'information. Cependant, les usagers vont devoir reformuler eux-mêmes leurs requêtes. Cette reformulation peut porter sur la structure logique de la requête (modification des opérateurs booléens) et/ou sur les concepts manipulés.
L'objectif de ce projet est d'aider l'utilisateur à retrouver les documents les plus pertinents par rapport à une requête donnée sans qu'il soit obligé à reformuler lui-même sa requête. Nous avons étudié pour cela le comportement des usagers afin de proposer de nouvelles techniques permettant d'améliorer la consultation d'une large base de données.
Les méthodes de reformulation de requêtes que nous proposons sont essentiellement cycliques et visent, dans notre cas, l'optimisation progressive et supervisée d'une requête finale. Ces deux méthodes sont assez semblables, mais leurs domaines d'application diffèrent car l'AFC sera plutôt réservée à l'analyse de petites entités de données (quelques documents de référence et donc quelques

blocs), le tri de matrices, pour sa part, conviendra pour des collections plus larges (quelques dizaines de documents de référence et plus de blocs sémantiques à traiter).
Les expérimentations que nous avons réalisées confirment l'intérêt de notre méthode. Le bruit engendré par ce type d'équation est inférieur à 8%, et si l'on se rapporte à la théorie du traitement du signal [15], ce rapport signal/bruit est plus qu'acceptable, et, de plus, contribue à l'extraction de signaux faibles souvent noyés dans la masse d'information et difficilement détectables a priori par l'utilisateur.
En un mot, la reformulation du besoin d'information est un des éléments clés pour obtenir des résultats pertinents dans un processus de recherche et notamment si l'on vise l'exhaustivité (minimiser le silence). Cependant, pour que la reformulation fonctionne correctement, nous devons enrichir ou affiner l'expression du besoin de l'utilisateur et pour cela l'assister dans le choix de la terminologie la mieux adaptée à sa recherche. Si la formulation des requêtes est une source d'erreurs, les systèmes retournent des résultats non pertinents car trop éloignés du sujet de la recherche ou, plus gravement, conduisent à la disparition de résultats pertinents. Nous proposons donc, dans un premier temps, de concevoir et de mettre en œuvre de meilleurs outils de recherche pour faciliter la création de requêtes efficaces mêmes si celles-ci doivent être plus complexes, inconvénient en partie compensé par leur réécriture automatique et bientôt par l'interrogation automatique des systèmes de recherche d'information les plus utilisés.

6 Bibliographie

[1] *Contraintes principales de la recherche d'information par mot clé*, Synomia Search, 2003

[2] Suy I. & Lang S. D.: *A competition-Based Connexionniste Model for Information Retrieval*, Intelligent Multimedia Information Systems and Management (RIAO), New York, 1994

[3] Mothe J.: *Modèle Connexionniste pour la recherche d'informations, Expansion dirigée de requêtes et apprentissage*, Thèse de Doctorat de l'Université Paul Sabatier, N°1839, Toulouse III, 1994

[4] Rocchio J. J.: *Relevance Feedback in Information Retrieval*, in Salton G. Editor, The Smart Retrieval System-Experiments in Automatic Document Processing, Prentice-Hall, Inc. Englewood Cliffs, NJ, 1971

[5] Salton G., Buckley C.: *Improving Retrieval Performance by Relevance Feedback*, Journal of the American Society for Information Science, 1990

[6] Robertson S. E., Spack-Jones J. K.: *Relevance weighting of search terms*, Journal of the American Society for Information Science, 1976

[7] Croft B.: *Experiments with representations in a document retrieval system*, Information Technology: Research and Development, 1983

[8] Harman D.: *Relevance Feedback Revisited*, In Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1992

[9] Wilkinson R. & Hingston P.: *Using The cosine Measure in A Neural Network for Document Retrieval*, Conference on Research and Development in Information Retrieval (SIGIR), Chicago (USA), 1991

[10] Kwok K. L.: *A Network Approach to Probabilistic Information Retrieval*, ACM Transactions on Information Retrieval Systems, 1995

[11] Boughanem M., Chrisment C., Mothe J., SouleDupuy C., Tamine L.*: Connectionist and genetic approaches to achieve IR*, In Soft Computing in Information Retrieval Techniques and Applications Editorial, 2000

[12] Chen H. & NG T.: *An algorithmic Approach to Concept Exploration in Large Knowledge Network (automatic thesaurus consultation),* Symbolic Branch and Bound Search vs Connexionniste Hop field net Activation Journal of the American Society for Informations Science, 1995

[13] Yang J., Korfhage R.: *Effects of Query term weights modification in document retrieval: A study based on a genetic algorithm*, In Proceedings of the Second Annual Symposium on Document Analysis and Information retrieval, 1993

[14] Tamine L.: *les Systèmes de Recherche d'Information : Reformulation de Requête et Apprentissage basé sur les Algorithmes Génétiques*, Thèse de Magister en Informatique, Université Mouloud Mammeri de Tizi-Ouzou, 1998

[15] MAX J.: *Méthodes et Techniques du traitement du signal*, 2ème édition, Masson, 1977