الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 –Guelma-

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de fin d'études de Master

Filière : Informatique

Option : Système d'informatique

Thème :

Reformulation automatique des requêtes pour améliorer la recherche d'information sur le web

Encadré Par :

Dr.Boughareb Djalila

Présenté Par :

Benhayaoum Khawla

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 –Guelma-

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de fin d'études de Master

Filière : Informatique

Option : Système d'informatique

Thème :

Reformulation automatique des requêtes pour améliorer la recherche d'information sur le web

Encadré Par :

Dr.Boughareb Djalila

Présenté Par :

Benhayaoum Khawla

Remerciement

Je remercie tout d'bord notre dieu qui j'a donné la force et la volonté pour élaborer ce travail.

J'adresse nos vifs remerciements à mon encadreur Mme, Boughareb Djalila pour le suivi de mon travail, pour ses lectures et pour sa grande patience.

Je remercie vont également aux membres de jury pour j'avoir honorés par leur participation à l'évaluation de ce modeste travail.

Notre reconnaissance aussi à tous ceux qui ont collaboré à mon formation en particulier les enseignants du département d'informatique, de l'université 08 mai1945Guelma.

Dédicace

Au bon DIEU le tout puissant pour qu'il m'éclaire le bon et droit chemin, je dédie ce Modeste travail à :

Ce qui est le plus cher et proche à mon cœur , source et symbole d'amour :

Ma mère qui a sacrifiée sa vie pour me rendre heureuse, que DIEU le tout puissant lui accorde une longue vie pleine de bonheur et santé.

A la mémoire de mon(feu)Père « Smain» que Dieu ai son Ame dans son vaste Paradis.

A mes frères et sœur : AbdelKader(sa femme Ahlem et ses enfant Djana,Isamil),Messaoud et Rima

Pour leur encouragement,que DIEU les gardes.

A mon fiancé M .amine .

Mes chères cousines :shayma,asma,manel.

Et à tous mes collegues de promotion M2SIQ

2017-2018

khawla

Résumé

Beaucoup d'études et statistiques sur le trafic de recherche sur le web ont confirmé que les utilisateurs tendent à l'emploi de requêtes courtes et ambiguës pour exprimer leur besoin en information. Ce qui conduit souvent au non satisfaction de ce besoin.

Une part importante du sens ou de la compréhension d'un terme doit être intimement liée à sa relation avec d'autres termes.

Partant de là, nous proposons un système qui permet de suggérer à l'utilisateur des termes pouvant l'aider à mieux exprimer son besoin en information, et d'améliorer ainsi la qualité des résultats fournis par le système de recherche.

Notre travail consiste à réaliser et développer un système de reformulation automatique des requêtes en utilisent la méthode pseudo réinjection de pertinence (PRF).

Mot clé : expansion de requêtes, moteur de recherche, recherche d'information textuelle.

Abstract

Many studies and statistics on web search traffic have confirmed that users tend to use short and ambiguous queries to express their need for information. This often leads to the non-satisfaction of this need. A significant part of the meaning or understanding of a term must be closely linked to its relation to other terms.

From there, we propose a system that allows to suggest to the user terms that can help him better expresses his need for information technology, and thus improve the quality of the results provided by the search system.

Our job is to realize and develop a system for automatic reformulation of queries by using the method of pseudo reinjection of relevance (PRF).

Keyword: query expansion, search engine, texte-based information retrieval.

الملخص

أكدت العديد من الدراسات والإحصاءات حول حركة بحث الويب أن المستخدمين يميلون إلى استخدام طلبات بحث قصيرة وغامضة للتعبير عن حاجتهم للمعلومات. هذا غالبا ما يؤدي إلى عدم تلبية هذه الحاجة.

يجب أن يرتبط جزء كبير من معنى أو فهم المصطلح ارتباطًا وثيقًا بعلاقته بمصطلحات أخرى.

ومن هناك، نقترح نظامًا يسمح باقتراح شروط المستخدم التي يمكن أن تساعده على التعبير عن حاجته للمعلومات بشكل أفضل، وبالتالي تحسين جودة النتائج التي يوفرها نظام البحث.

مهمتنا هي تحقيق وتطوير نظام لإعادة صياغة استعلامات التلقائي عن طريق استخدام طريقة الصلة .pseudo réinjection (PRF)

الكلمة الرئيسية: توسيع طلب البحث، محرك البحث، البحث عن المعلومات النصية.

Sommaire

Liste des figures

Abréviation et acronymes

RI : Recherche d'information.

SRI : le système de recherche d'information .

WSD : Word sens désambigüisation.

PRF : Pseudo Relevance Feedback.

SMO : Sequentiel Minimal Optimization.

QE : Query Expansion.

WE : Word Embeddings.

P : précision.

NDCG :normalized discounted cumulative gain.

MRR : Metal Removal Rate.

Introduction générale

Aujourd'hui, internet est devenu très vaste et héberge des millions d'informations de différents types, c'est un environnement d'information où vous pouvez chercher des informations dans n'importe quel domaine, il représente le moyen d'atteindre votre objectif.

La recherche d'information est devenue plus facile et rapide, sans effort, car l'information vous vient avec un simple clic sur un bouton, c'est le principe de base d'un système de recherche d'informations (SRI). Un SRI peut trouver des documents non organisés dans un grand espace, selon le besoin d'information de l'utilisateur dont le type de recherche le plus connu recherche par "mots-clés".

Mais en revanche, il est très difficile de savoir si cette information correspond ou non au besoin de l'utilisateur, étant donné l'énorme quantité d'informations.

En plus, le mot-clé utilisé par l'utilisateur est censé représenter une partie du contenu du document et de la requête, en tenant compte de la manipulation représentative et simple des mots-clés du contenu. Pour cette raison, l'expansion de la requête est nécessaire car elle tente d'ajouter d'autres termes à la requête pouvant aider à se rapprocher du besoin en information de l'utilisateur. En identifiant les requêtes en fonction des similitudes entre le document et la requête, nous avons la possibilité d'identifier des documents plus pertinents avec une requête étendue.

Objectifs

L'expansion de la requête permet d'ajouter de nouveaux termes à la requête initiale de l'utilisateur qui est le plus souvent courte et ambigüe permettant ainsi de formuler une requête étendue plus expressive que la requête initiale.

Dans ce travail, l'objectif est de réaliser un système d'expansion des requêtes en se basant sur la technique de la pseudo-réinjection de pertinence qui est proposé par Belkin.

Pour cela, notre système permet à un utilisateur de faire une recherche par mot-clé puis il traite cette requête initiale et retourne des suggestions d'expansion de cette requête par d'autres mots-clés. L'utilisateur évalue finalement les mots-clés d'expansion retournés par le système.

Organisation du mémoire

Ce mémoire est organisé en 4 chapitres comme suit :

Le premier chapitre, expose un état de l'art sur la reformulation des requêtes, nous avons présenté ces approches par classes.

Le deuxième chapitre, explique notre conception de notre application ainsi que la description détaillée de notre système.

Dans le troisième chapitre, nous décrivons les étapes d'implémentation et les outils utilisés pour réaliser notre système ainsi que les résultats obtenus.

Enfin, nous clôturons ce mémoire par une conclusion générale.

Chapitre 1 :

Un aperçu des déférentes techniques de reformulation de requêtes

Chapitre 1 : un aperçu des déférentes techniques de reformulation de la requêtes

1. Introduction

Pour effectuer une recherche sur le web, l'utilisateur doit choisir des bons termes qui expriment son besoin pour accéder à l'information qu'il désire. Les utilisateurs tendent d'utiliser des requêtes courtes exprimées par des termes ayant différentes significations. Le résultat généré par ces requêtes ne satisfait pas toujours le besoin de l'utilisateur. Ces requêtes courtes qui visent des domaines d'intérêts différents sont appelées requêtes ambigües.

Afin de résoudre ce problème, la reformulation de requête se présente comme une solution permettant d'aider l'utilisateur à accéder à l'information désirée, ceci par la modification de la requête initiale afin de lever l'ambiguïté, et d'ajouter des termes plus significatifs [RUT03; FON05; VOO06; SON07; KAN08; WAN09; LV10].

2. Définition

2.1. Ambiguïté :

C'est un phénomène traditionnel dans le langage naturel qui est dû à la polysémie [Boughareb 2014]. La polysémie de mot est la caractéristique d'un mot ou d'une expression à avoir des sens différents selon le contexte dans lequel ils apparaissent [Boughareb 2014]. En recherche d'information (RI), une requête ambiguë génère des résultats variés et souvent hétérogènes. Par exemple le mot « pascal » désigne un langage de programmation, le nom d'une personne, le nom d'un établissement, d'une station de radio, etc. Et là, la conduite d'une recherche sur Google par le biais de la requête « pascal » génère des pages sur des domaines différents, et l'utilisateur qui conduit cette recherche ne sera pas souvent satisfait du résultat retourné.

2.2. Reformulation de requête :

La reformulation de requête est une technique qui consiste soit à modifier la requête initiale par l'ajout de nouveaux mots clés afin de produire une nouvelle requête spécifique et là on parle de l'expansion de requêtes ou par la repondération des poids des termes de cette requête [BAE99; RUT03; FON05; BIA09]. Le but de cette reformulation est d'adapter la requête avec le besoin de l'utilisateur. Dans ce qui reste on discutera des approches d'expansion de requêtes.

3. Classification des approches d'expansion de requêtes

3.1. Selon le degré d'implication de l'utilisateur :

Selon le degré d'implication de l'utilisateur dans le processus d'expansion de requêtes, on a deux approches d'expansion, une approche basée sur un processus interactif et une autre basée sur un processus automatique.

3.1.1. Approche Interactive :

Selon cette approche, l'utilisateur choisit les mots clés d'expansion parmi une liste que le système suggère. Cette approche donne de bons résultats mais elle nécessite de l'expertise, et en même temps elle permet à l'utilisateur de contrôler le traitement de requête.

3.1.2. Approche Automatique :

Dans cette approche d'expansion, l'utilisateur ne peut pas avoir accès à la requête, l'expansion est donc faite de manière automatique par le système de recherche d'information (SRI) soit sur la base des ressources linguistiques ou bien des documents.

3.2. Selon la source des termes d'expansion :

3.2.1. Méthode basée sur la réinjection de pertinence :

Selon Rochio [ROC71], la réinjection de pertinence est un processus où le SRI fournit à l'utilisateur un ensemble de documents comme réponse à sa requête. L'utilisateur sélectionne ensuite les documents qui correspondent le mieux à son besoin en information. Ensuite, et sur la base de cet ensemble d'interactions, le SRI peut effectuer une recherche de documents plus raffinée afin de fournir plus de résultats pertinents [BOU2014].

Le feedback permet à l'utilisateur d'extraire les termes d'expansions par les pages consultés et leur pertinence.

L'inconvénient majeur de cette méthode est que la qualité de reformulation dépend fortement de l'aptitude des utilisateurs à donner des jugements corrects de la pertinence des documents [BOU2014].

3.2.2. Méthode basée sur le pseudo réinjection de pertinence :

Cette méthode considère que les k premiers documents récupérés par une requête comme pertinents. Ces documents sont ensuite utilisés comme un feedback de pertinence ce qui rend le processus de recherche plus rapide. Cette méthode n'est pas totalement précise parce que les k premiers document ne sont pas tous pertinents. Pour résoudre ce problème,

Belkin [BEL00] a suggéré l'évaluation des documents initiaux avant l'extraction des termes d'expansion des documents.

3.2.3. Méthode Basée sur les ressources Sémantiques :

Elle est basée sur la sémantique pour déterminer le sens d'un mot à travers le contexte computationnel [NAV09]. Les ressources terminologiques les plus utilisées sont les thesaurus et les ontologies, où Wordnet représente la ressource la plus utilisée. En effet, cette méthode dépend des techniques de désambigüisation de sens des mots (Word sens désambigüisation) ou encore WSD.

Dans [VOO94], l'auteur propose d'enrichir la requête par des synonymes extrait depuis Wordnet. Tandis que dans le travail de Navigli et Velardi [NAV05] d'autres critères sont exploités comme les hyperonymes, et les hyponymes des mots. Le problème majeur avec ces approches est lié à l'aspect générique de WordNet.

Pour cela, les approches basée ontologie de domaine Fournit des solutions pour surmonter le problème qui se produit à wordnet.

3.3. Selon le principe de génération des termes d'expansion :

On a trois groupes principaux sont :

3.3.1. L'approche Linguistique :

Elle s'intéresse à la découverte des relations syntaxiques, lexicales et sémantiques entre les mots en s'appuyant sur des ressources terminologiques telles que les thesaurus et les ontologies [VOO93, 94; BHO07; SEG14].

3.3.2. L'approche Statistique :

Elle est basée sur le principe de l'interconnexion des termes à travers l'exploitation de la technique de cooccurrence de termes, et ceci dans un contexte global quand elle est appliquée à tout le corpus de documents du système ou local dans le cas où elle est appliquée aux premiers documents résultants de l'exécution de la requête.

3.3.3. L'approche Mixte :

Cette approche combine les deux approches précédentes. Elle repose sur l'analyse de la distribution des mots en bénéficiant d'une ressource terminologique.

Dans les travaux de Liu et al. et de Fang qui ont utilisé Wordnet pour désambigüiser les termes de la requête, il a été prouvé que l'approche mixte produise des résultats meilleurs que les autres approches.

4. Processus d'expansion

Il comprend deux étapes intimement liées : le prétraitement de données et la sélection des candidats termes.

4.1 Prétraitement de données :

Cette étape est généralement indépendante de la requête de l'utilisateur, elle dépend du type de source de données. Elle consiste à transformer la source de données brutes utilisées pour l'expansion des requêtes en un format qui peut être traité plus efficacement par la suite.

4.1.1 Collection de documents :

Comme dit plus haut, les informations qui existent dans les documents les mieux cités elles présentent la réponse à la requête comme une source d'expansion très adoptée. Le traitement de ce type d'information basé sur le processus d'indexation pour la pertinence de requête qui permet de produire une représentation terminologique pour chaque document [CAR01];[BAI05]; [VOO04]; [DIA06]; [CHI07]. Les termes d'expansion sont sélectionnés depuis la terminologie créée.

4.1.2. Les textes d'ancrage :

Pour faire un traitement sur cette source il faut analyser les collections d'hyperliens pour extraire les textes balises d'ancrages [KRA04]. Après, ce texte va subir le prétraitement pour extraire les termes expansion. D'après les expérimentations qui confirmé la technique de reconnaissance de textes les balises d'ancrages surmonte la technique similaires qui basée sur la fréquence d'un mot dans le texte des documents.

4.1.3. Les fichiers logs :

Les fichiers logs Sont des fichiers textes qui sauvegardent l'ensemble de données d'un système informatique dans une forme en générale datés et classée en ordre chronologique. Ces fichiers contient tout les ensembles de requêtes exécuté et ainsi les documents qui visité par l'utilisateur pendant navigation. L'analyse des fichiers permet d'éliminer les pages non pertinentes et identifier les pages pertinentes à une requête donnée. Alors, l'extraction des termes sémantiquement lié au requête peut possible. [BEE00; CUI03; BIL03, BOU13b].

4.2. Sélection des candidats termes :

La sélection des candidats termes consiste à partir l'extraction des données traitée

Précédemment qui elle peut aider pour l'enrichisse la requête initiale. Pour choisir la liste de terme approprié à l'expansion de requête, il faut appliquer la phase de désambigüisation pour définir le sens. On peut effectuer l'ajout des termes soit par l'expansion de chaque requête aux autres termes dans façon séparable ou façon inséparable où la requête on peut exprimer comme un lot de termes.

Pour rapprocher l'information au l'utilisateur, il obligerait d'appliquer une technique qui sélectionné les termes d'expansion à partir la ressource collecté. D'après la technique de tri de Hartmann [HAR92] qui contient la formule de pondération par Robertson et Sparck-Jones [ROB76], il représente en le choix des termes qui a les mêmes valeurs de poids dans documents pertinents et la faiblesse de probabilité d'apparition dans les documents non pertinent.

La formule qui est présentée par l'équation a permis d'obtenir de bons résultats de reformulation.

$$W_{ij} = \log_2 \frac{p_{ij}(1 - q_{ij})}{q_{ij}(1 - p_{ij})}$$

Où Wij représente le poids du terme i dans la requête j, $Pij(1 - Qij)$ est la probabilité que le terme i apparaisse dans les documents pertinents pour la requête j et $Qij(1 - Pij)$ mesure la probabilité que le terme i apparaisse dans les documents non pertinents pour la requête j.

D'autres phases de tri est basé sur le principe d'augmentation des mots [BOU99, 00 ; BAI06; BOU13b] où les termes utilisé pour ajouter l'enrichesse au les requêtes initiales. Par exemple, dans le couple de ce mot (équation, traitement) chacun sert d'un contexte à l'autre qui permet de contraindre les termes connexes. Dans ce cas, les mots « médicament» et « thérapie » auront une probabilité de cooccurrence beaucoup plus importante avec « équation, traitement » que le mot « mathématique ».

4.3. Reformulation des requêtes :

Le besoin de reformulation en information consiste à redéfinir le besoin d'utilisateur au fur et à mesure de la session de recherche. Cette phase peut être faite en différentes manières :

- Manuellement, dans le cas où l'utilisateur soumet lui-même une nouvelle requête.

➢ Automatiquement, tant que le SRI consiste sur les termes importants dans les documents les plus pertinents ou visités par l'utilisateur, qui sont réutilisés.

Suppose que dans l'approche automatique l'utilisateur avance une requête au SRI pour récupérer les documents pertinents, sinon cette requête incombait être ré-écrite ou reformulée pour récupérer plus de documents pertinents. La Figure 1 donne un aperçu du principe de la reformulation de requête dans le cadre d'un SRI [EYA16].

L'objectif de processus de reformulation de requête est pour générer une nouvelle requête plus pertinence pour obtenir un ensemble de résultats plus pertinents, d'après connu le domaine cible, en utilisant les termes clés qui reçu dans les documents. La requête initiale est formulée par l'utilisateur, la modification de cette requête peut se faire soit par réinjection de pertinence (relevance feedback) [SAL97], soit par expansion de requêtes (query expansion) [EFTH96] [EYA16].

Il ya deux étapes fondamaux pour faire la reformulation de la requête sont :

i. trouver des termes d'extension à la requête initiale, et
ii. ré-pondérer les termes dans la nouvelle requête.

La base de la stratégie d'expansion de requête est de comparer le contenu de la requête avec les documents de la collection dans une façon simple. Où tout des documents pertinents sera souvent incomplet. Des travaux de recherche qui proposent l'ajout d'autres des termes existe dans les documents pertinents ou l'ajout des termes sémantique proches ou aussi l'ajout des termes voisins avec l'utilisation des calculs de poids de similarité entre termes [EYA16].

Dans la littérature a proposée des Différentes méthodes d'expansion de requêtes [XU96, ADR99, BAZIZ03, LATIRI12, CARP12, NAWAB16].

Le but de ces méthodes d'expansion est incrémenter le nombre de documents pertinents qui retrouvés et aussi perfecter le classement des documents les plus pertinents [EYA16].

Dans le cadre de la recherche d'information, il y a un obstacle pour retourner les résultats celui est entre les termes des documents et termes des requêtes qui est la barrière de la non compatibilité et appelé aussi "term mismatch"[EYA16]. Cela est connu comme étant le problème de vocabulaire [FUR87], amplifié par les synonymes (mots différents avec le même sens comme "java"), la polysémie (différents termes avec le même sens, comme "tv" et "télévision"). Les synonymes qui contiennent des mots à partir des mêmes mots (comme pour les formes au pluriel "télévision" et "télévisions") peuvent mener à un échec pour récupérer les documents pertinents, avec une diminution pour rappeler ces documents (la capacité du

système à retourner tous les documents pertinents en réponse à la requête). En plus de ca, la polysémie est la source de la récupération de documents erronés et non pertinents, tout cela signifie qu'une diminution dans précision des résultats (la capacité de retourner uniquement les documents pertinents) [EYA16].

Pour résoudre au problème de vocabulaire, il y a plusieurs approches qui ont proposées, comme les raffinements interactifs de requêtes, la reformulation de requêtes par réinjection de la pertinence, la désambigüisation des sens de mots et le clustering des résultats de recherche.

Il y a une approche qui est le plus naturelle et succès qui les autres approches est la technique d'expansion ou de reformulation des requêtes initiales avec autres termes qui représentent au mieux l'intention des utilisateurs, ou dans une façon plus simple produire une requête plus utile et plus susceptible pour récupérer des documents pertinents.

Pendant les dernières décennies, un grand nombre de techniques d'expansion automatique de requêtes est représenté par l'utilisation d'une Divers approches qui se consistent à plusieurs sources de données et utilisent des méthodes sophistiquées pour trouver de nouvelles fonctionnalités en corrélation avec les termes de la requête [MIT98; CARP02; LIU04; LEE08; LATIRI12]. Ces contributions ont montré à travers les études expérimentales que les résultats de l'expansion automatique de la requête permettent de donner des résultats plus pertinents avec des améliorations de 10% et plus [EYA16].

Dans [CARP12], les auteurs ont fait une large étude comparative des approches qui concernent l'expansion automatique des requêtes. Ils ont montré pourtant la continuité du problème de vocabulaire dans certains travaux, l'expansion automatique de requêtes a le potentiel de surmonter le problème majeur des SRI, à savoir :

La difficulté pour fournir une description plus précise de besoin en information aux utilisateurs [EYA16].

Dans [Huang09], les auteurs ont analysé et évalué différentes stratégies de reformulation de requêtes à partir des fichiers logs du web. à partir de ce analyse, ils ont conclu que les stratégies de reformulation ont différentes caractéristiques, et le plus effectives sont ajout/suppression de mots, substitution de mots, expansion avec acronymes et la correction orthographique[EYA16].

D'autres travaux se sont intéressés à l'expansion sémantique des requêtes, Afin de surmonter le problème de dérive sémantique (semantic mismatch)[PACKER12; CURé13]. En réalité, les termes d'expansion qui ont liaison sémantique avec les termes initiaux de

requêtes rajoutent plus d'explicitation au contexte de la recherche et améliore ainsi il est pertinence des résultats. Différents domaines ont basé sur l'expansion sémantique, par exemple la RI sociale, la RI biomédicale, etc. Ils utilisent les concepts issus des terminologies et des thésaurus ou encore les relations sémantiques entre les termes ou concepts pour donner une mieux représentation de thème(le topic de la requête).

En outre, la reformulation par retour de pertinence propose de formuler la requête initiale pour commencer à la recherche d'information, alors on fait une modification dans façon itérative par des jugements de pertinence et/ou de non pertinence de l'utilisateur pour d'ajuster la requête par expansion, repondération ou combinaison des deux procédures, en attendant le résultat de la recherche soit satisfaisant.

Dans [CARP12], selon les auteurs, ils ont donné une classification des approches et techniques d'expansion automatique de requêtes en cinq groupes d'après le paradigme conceptuel utilisé afin de trouver les caractéristiques d'expansion, sont: méthodes linguistiques, approches statistiques spécifiques au contexte, approches statistiques spécifiques aux requêtes, analyse des fichiers de log et les données du web.

Dans la reformulation de requête, il y a plusieurs des techniques et nous va détailler deux principales ces techniques, sont :

i. reformulation par réinjection de la pertinence (relevance feedback),

ii. Reformulation par pseudo-réinjection de la pertinence

(Pseudo relevance feedback ou blind query expansion) [ROC71].

4.3.1 Reformulation par réinjection de la pertinence :

La réinjection de la pertinence est une technique qui utilise pour améliorer la performance de la recherche d'information [ROC71; SAL97]. Pendant ce processus, l'utilisateur utilise une requête initiale, puis servit un retour sur la pertinence des documents. Les termes de ces documents (jugés pertinents) sont donc ajoutés à la requête initiale. Le but de la technique de reformulation par réinjection de la pertinence est améliorer la qualité de recherche car est la seule évaluation de la similarité entre les requêtes et les documents n'est plus suffisante. On résume cette techniques sur quatre étapes, sont :

i. Les utilisateurs effectuent une première requête.

ii. Des documents sont retournés en fonction de cette première interrogation.

iii. Les utilisateurs doivent ensuite indiquer parmi les documents retournés, lesquels sont pertinents, et/ou lesquels ne le sont pas.

iv. La requête de départ est alors modifiée automatiquement pour tenir compte des jugements des utilisateurs [EYA16].

La méthode de la réinjection de la pertinence a été utilisée dans différents zones de recherche, intégrée dans des SRI [KWAN15], utilisée dans le cadre de la RI d'image [DUAN16] ou aussi pour la recherche de vidéos [FERN16]. Cette méthode par rapport aux techniques standards de recherche, elle a montré une amélioration de performance [EYA16].

4.3.2 Reformulation par pseudo-réinjection de la pertinence :

Dans la reformulation par pseudo-réinjection de la pertinence (Blind Feedback ou encore Pseudo Relevance Feedback, notée PRF) on utilise les techniques de réinjection automatique à l'aveugle afin de construire la nouvelle requête. L'essentielle idée de la PRF est basée sur l'hypothèse que les premiers documents pertinents contiennent de nombreux termes valables qui aident à distinguer les documents pertinents des non pertinents. Généralement, les termes d'expansion sont extraire à partir de leur distribution dans les documents retournés, ou selon la comparaison entre la distribution de termes dans les documents retournés et l'ensemble de documents de la collection. Plusieurs autres critères ont été proposés par exemple idf [ROC71]. De plus, la PRF est une technique couramment utilisée pour faire face à l'explosion de l'information sur le web pour améliorer la performance de recherche [BUCK92; YU03]. L'utilisation de Pseudo-Relevance Feedback a fait l'objet d'un grand nombre d'études depuis plusieurs décennies et mettre plusieurs modèles dans ce cadre. [THES14; MIN10; HAMM13]. Pour déterminer les termes d'expansion, les auteurs additionnent les poids des relations d'un terme candidat avec chacun des termes de la requête. S'il est fortement en

Relation avec les termes de la requête et le choix des termes candidats [HAMM13]. Dans le cadre de modèle de langue, Ils ont intégré la technique de PRF. Dans le même cadre du modèle de langue, dans une étude plus récente [HAZI15], les auteurs ont fait une analyse sur des méthodes de lissage dans les modèles de langue pour la PRF. En plus, [LI12] la technique de PRF utilise afin d'estimer la difficulté des requêtes qui permet d'estimer la performance de la recherche pour les requêtes de recherche d'images. Le Tableau 1 montre quelques fonctions de classement de termes basées sur la distribution des termes dans les documents de pseudo-réinjection de la pertinence [EYA16].

La notation dans le Tableau 1 est comme suit :

i. t est un terme ;

ii. w(t, d) indique le poids du terme t dans le document de pseudo-réinjection de pertinence d ;

iii. p (t|R) et p (t|C) représentent respectivement la probabilité d'occurrence du terme t dans les documents de la pseudo-réinjection de pertinence R ainsi que dans toute la collection de documents C.

Référence	Fonction	Forme mathématique
(Rocchio, 1971)	Poids de Rocchio	$\sum_{d \in R} w(t,d)$
(Robertson et Sparck Jones, 1988)	Modèle Indépendant Binaire	$log\frac{p(t\|R)[1-p(t\|C)]}{p(t\|C)[1-p(t\|R)]}$
(Doszkocs, 1979)	Chi-square	$\frac{[p(t\|R)-p(t\|C)]^2}{p(t\|C)}$
(Robertson, 1991)	Robertson selection value (RSV)	$\sum_d w(t,d).[p(t\|R) - p(t\|C)]$
(Carpineto *et al.*, 2001)	Kullback-Leibler distance (KLD)	$p(t\|R).log\frac{p(t\|R)}{p(t\|C)}$

Tableau 1 : Fonctions principales de classement de termes basés sur l'analyse des distributions des termes dans les documents de pseudo-réinjection de pertinence [CARP12].

5. Les travaux connexes

Il y a beaucoup de travaux visant à améliorer les requêtes de recherche ici nous intéressons à ceux basés sur la méthode de pseudo-réinjection de pertinence. Dans le travail de [ARO 17] les chercheurs examinent l'utilisation de méthodes d'expansion de requête (QE) dans l'extraction de phrases pour des requêtes non-factoid afin de résoudre le problème de correspondance entre les termes des documents et ceux des requêtes.

Deux approches alternatives QE: i) le pseudo rétroaction de la pertinence (PRF), en utilisant la sélection de termes Robertson [ROB90], et ii) l'expansion de mots (WE) de la requête, sont explorés. Des expériences sont réalisées sur l'ensemble de données WebAP développé à l'aide de la collection TREC GOV2.

Les résultats expérimentaux obtenus avec la précision à 10 P @ 10, et NDCG @ 10 (normalized discounted cumulative gain) et MRR (Metal Removal Rate) montrent que l'amélioration de la qualité des résultats obtenus avec les PRF est statistiquement significative par rapport aux modèles de base [ARO17]. Dans le travail de [SPA11], les auteurs proposent une nouvelle méthode de génération de requêtes utilisant des informations spatiales, temporelles et textuelles basées sur un pseudo retour de pertinence. La méthode proposée génère de nouvelles requêtes spatio-temporelles à partir des résultats de recherche initiaux.

En utilisant ces requêtes, les résultats de la recherche sont réécrits de sorte que plus de résultats associés obtiennent un rang plus élevé.

Les résultats expérimentaux montrent que la méthode proposée surpasse une méthode de référence lorsque les cibles de recherche n'ont pas d'informations textuelles riches.

Aussi, Dans le domaine de la recherche d'informations médicales qui joue un rôle de plus en plus important pour aider les médecins et les experts du domaine à mieux accéder aux connaissances et informations médicales, et à soutenir la prise de décision. [HAO17] ont proposé un nouveau système de recherche d'information médicale avec une stratégie d'expansion de requête en deux étapes, qui est capable de modéliser et d'incorporer efficacement les associations sémantiques latentes pour améliorer la performance.

Ce système se compose de deux parties. Premièrement, les chercheurs ont appliqué une approche heuristique pour améliorer la méthode de pseudo réinjection de pertinence en élargissant itérativement les requêtes pour augmenter le score de similarité entre les requêtes et les documents. Deuxièmement, pour améliorer la performance de récupération avec des bases de connaissances structurées, nous avons présenté un modèle de pertinence sémantique latent basé sur la factorisation tensorielle pour identifier les modèles d'association sémantique dans des contextes clairsemés.

Ces modèles identifiés sont ensuite utilisés comme chemins d'inférence pour déclencher l'expansion de la requête basée sur la connaissance dans la recherche d'informations médicales. Les expériences menées sur l'ensemble de données TREC CDS 2014:

1) a montré que la performance du système proposé est nettement meilleure que celle du système de base et des systèmes rapportés lors de la conférence TREC CDS 2014, et qu'elle est comparable aux systèmes de pointe.

2) a démontré la capacité de méthodes d'enrichissement sémantique à base de tenseurs pour les tâches de recherche d'information médicale [HAO17].

Pour améliorer la recherche d'événements en vidéo ont proposé une nouvelle méthode nommée MultiModal Pseudo Relevance Feedback (MMPRF), qui exploite non seulement des caractéristiques sémantiques, mais aussi des caractéristiques de bas niveau non sémantiques pour la recherche d'événements en l'absence de données d'apprentissage.

Evaluée sur l'ensemble de données TRECVID MEDT, l'approche améliore la référence jusqu'à 158% en termes de précision moyenne.

6. Conclusion

Chapitre 1 : un aperçu des déférentes techniques de reformulation de la requêtes

Dans ce chapitre, on a défini l'ambigüité et la reformulation des requêtes on peut dire l'expansion de requête.

Et aussi parle sur les différentes classifications d'approches d'expansion ces requêtes qui sont trois critères principaux :

Selon le degré d'implication d'utilisateur, on a deux approches basé sur le processus interactive où l'utilisateur est un acteur directe et cette approche donne les meilleur résultats et l'autre approche basé sur le processus automatique où l'utilisateur ne peut pas intervenue directe.

Selon la source des termes d'expansion, on a la méthode basée sur la réinjection de pertinence, et autre basée sur la pseudo-réinjection de pertinence et l'autre basée sur les ressources sémantiques.

Selon le principe de génération des termes d'expansion, on a l'approche linguistique est basée sur la ressource terminologique, l'approche statistique est basée sur la découverte des relations statistiques et en dernier l'approche mixte est basée sur les deux approches précédent.

Comme nous discuté les étapes du processus d'expansion qui est le prétraitement de données et la sélection des candidats termes.

Chapitre 2 :

Conception du système

Chapitre 2 : Conception du système

1. Introduction

Notre application consiste à concevoir un système de reformulation automatiques des requêtes pour une future utilisation afin d'optimiser la recherche sur le web.

Le système consiste tout d'abord d'associer à chaque nouvelle requête la classe appropriée, après il sélectionne les requêtes les plus similaires à celle-ci au sein de la même classe. La classification des requêtes est faite en utilisant les SVM.

Dans ce chapitre nous avons présenté l'architecture générale du système et les différentes étapes de la conception.

2. L'objectif de notre système

L'objectif est l'expansion de requêtes courtes (qui comporte qu'un seul mot) dans le but de lever leur ambigüité en fait les mots de n'importe quelle langue peuvent avoir plusieurs sens selon le contexte où ils apparaissent par exemple le mot pascal peut signifier un langage de programmation, le nom d'une université, le nom d'un savant, d'une station de radio ou d'une chanteuse. Notre but est de proposer à l'utilisateur d'autres mots pouvant aider à mieux comprendre son besoin de recherche par exemple si l'utilisateur veut effectuer une recherche sur le langage pascal en utilisant la requête à mot unique « pascal » le système pourra lui aider dans sa recherche par proposer d'étendre la requête initiale avec le mot **langage,** la nouvelle requête obtenue après l'expansion sera **langage pascal.** Cette requête est beaucoup plus significative et elle permet de retourner des résultats de recherche plus adéquats au besoin informationnel de l'utilisateur.

3. Architecture générale du système

Le développement de ce système est réalisé en six grandes phases: la phase de collecte de données, la phase de prétraitement, la phase d'apprentissage la phase de classification, la phase d'expansion et la phase de test.

La figure (Figure 2.1*)* explique l'architecture générale du système

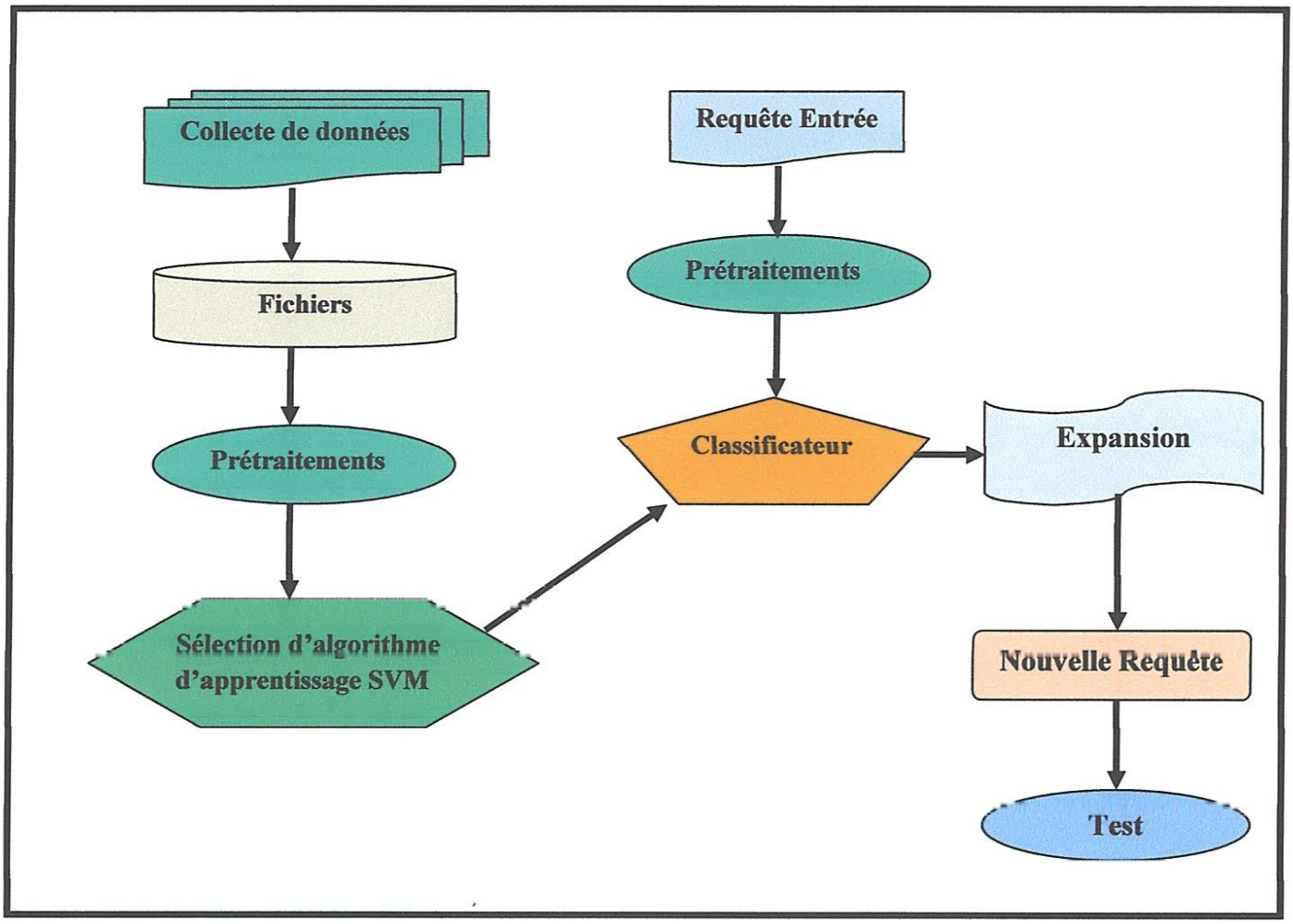


Figure 2.1 : *Architecture générale du système.*

1. La collecte de données

Pour collecter les données, nous avons utilisé le moteur de recherche Google.

On fait on a effectué plusieurs recherches par des requêtes avec un seul mot dans le domaine informatique, ensuite nous avons téléchargé la première page de résultats de chaque requête. Ce que nous avons obtenu est sauvegardé dans des fichiers texte (*.txt) et collecter dans un répertoire.

Dans l'exemple ci-dessous, nous verrons un échantillon des résultats de notre recherche liés au mot alogarithme.

Exemple :

Algorithms | Computer science | Computing | Khan Academy

https://www.khanacademy.org/computing/computer-science/algorithms

We've partnered with Dartmouth college professors Tom Cormen and Devin Balkcom to teach introductory computer science algorithms, including searching, sorting, recursion, and graph theory. ... Learn how to use asymptotic analysis to describe the efficiency of an algorithm, and how to ...

2. Prétraitement :

Cette étape comporte les sous étapes suivantes :

- ***Suppression des mots vides (Filtrage) :*** Les « stop-words » sont des mots qui n'apportent pas de sens lors de l'analyse lexicale d'un texte. Ce sont donc des mots que l'on exclut généralement lors de l'indexation ou de l'analyse d'un texte. C'est une liste de stop words en anglais comprenant des pronoms, des mots de liaison et quelques adverbes, noms [web2].

Cette étape permet de ne garder que les termes significatifs qui représentent le contenu de document.

Exemple :

Les mots vides sont : « the », « if », « actually », « after », « your », « in », « to ».

Exemple extrait de la collection :

Texte avant le filtrage :

« An application program (app or application for short) is a computer program designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user ».

Texte après le filtrage :

« Application program app application short computer program designed perform group coordinated functions tasks activities benefit user ».

- **Calcule de la cooccurrence des mots :**

La cooccurrence est la présence simultanée de deux ou de plusieurs mots (ou autres unités linguistiques) dans le même énoncé (la phrase, le paragraphe, l'extrait) [web6].

Cette étape permet de calculer combien de fois des pairs de mots extraits de la collection ont été apparus ensemble dans les phrases des documents de la collection.

Exemple :

Dans notre collection de documents nous présentons le paragraphe ci-dessous qui est le résumé d'un résultat en réponse à la requête « Exploitation »:

« Exploitation Meaning in the Cambridge English Dictionary

https://dictionary.cambridge.org/dictionary/english/exploitation

the act of using someone unfairly for your own advantage: Marx wrote about the exploitation of the workers. Treating people or animals badly. a raw deal idiom»

Si on compte le nombre d'occurrences du mot « **exploitation** » on aura le nombre trois (3) comme occurrence.

Ainsi qu'on peut citer quelque statistique sur notre collection (répertoire de documents)

Le mot « **exploitation** » existe **17 fois** sur toute la collection.

Il y a le mot « **meaning** » existe **3 fois** dans la collection.

La couple de mots « **exploitation, meaning** » apparait **3 fois** dans la toutes la collection.

3.Apprentissage :

L'apprentissage est un champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou problématiques par des moyens algorithmiques plus classiques [web7].

La première phase est l'entraînement, lors de laquelle les paramètres du système sont réglés sur une base d'apprentissage contenant des documents avec leurs classes respectives. Le système apprend l'association entre les documents et leurs classes [MAS].

La seconde phase est le test, que le système assigne une classe à chaque nouveau document entrant. Habituellement, les paramètres des systèmes d'apprentissage sont mis à jour périodiquement pendant un laps du temps où il n'y a pas de traitement à effectuer sur des documents arrivants [MAS].

Le SVM qui est un algorithme d'apprentissage automatique très efficace dans les problèmes de classification [web4]. On utilise cet outil de classification pour permettre d'estimer pour chaque requête entrée sa propre classe d'appartenance à partir de l'apprentissage supervisé préalablement effectué.

La génération du modèle de classification se fait en appelant la fonction «SMO» existante sous WEKA [UG].

- **Le Codage :**

Le codage est une façon de représenter un objet quelconque de façon à le pouvoir utiliser autrement, Le but d'attribuer un code pour chaque mot ou paire de mots est de faciliter leur classification par un algorithme de classification quelconque tel que l'algorithme **BayesNet, NaiveBayes Logistic** et **SMO** (l'algorithme que nous avons choisi) à noter seulement qu'il existe plein d'autre algorithme de classification.

Pour ceci on s'est penché sur le codage manuel de chaque mot sur 5 bits ainsi que les mots qui appartiennent à la même famille vont avoir un code qui peut varier dans une plage ou intervalle bien précise.

Exemple :

Le mot **Algorithm** a pour code la séquence **1.0003** parce que la classe **Algorithme** qui regroupe un ensemble de requêtes et de mots en rapport avec l'algorithmique a comme plage de valeurs **[1,0000-2,0000[**.

Pareillement la paire de mots **Algorithm meaning** faisant partie de la même classe a eu le code 1,0005 donc elle (la paire) figure dans la même plage et par conséquence l'Algorithme de classification SMO va la classer au voisinage du mot **Algorithm**.

4.Classification des requêtes :

A ce stade d'étude nous somme focalisé sur l'élaboration d'un modèle de classification obtenu lors de la phase d'apprentissage pour classer les nouvelles requêtes. Les requêtes vont subir les mêmes étapes du processus de prétraitement.

La requête codée va être attribuée par notre classifieur à la classe appropriée de façon automatique.

5. Expansion de Nouvelles Requêtes:

A ce stade et pour chaque nouvelle requête à mot unique saisie par l'utilisateur le système propose des mots pour l'étendre et la rendre plus significative par exemple si l'utilisateur tape la requête **Algorithm** le système et après avoir identifiée la classe de cette requête, il va proposer de l'étendre par le mot **definition** et la requête étendue sera **Algorithm definition.**

4. Description du projet

Dans cette partie, nous allons présenter les composants de notre système :

4.1 Package :

On a un seul « package » nommé **Reformulation** qui contient les classes et la bibliothèque qui englobe deux autres bibliothèques importées afin d'integere l'outil WEKA (pour les Algorithmes de classification) et UCANACCESS (pour établir un pont « Bridge » entre l'application et la base de donnée Access).

- **La base de données :** le but de création de cette base est pour enregistrer les données et ensuite les récupérer.

Cette base contient 5 tables :

1. Table Occurrence : cette table permet d'enregistrer les mots que nous avons recherchés (requêtes), leur nombre d'occurrence et les noms des fichiers dans lesquels ils apparaissent.

2. Table Codage : cette table permet d'enregistrer les mots, leurs codes, leurs nombres d'occurrences leurs classes d'appartenance.

3. Table OccurenceT : cette table comporte les mots et leur nombre d'occurrence dans toute la collection.

4. Table Classe : dans cette table nous avons 11 classes chaque famille a son intervalle de codage et les mots caractérisant cette classe.

5. Table Suggestion : dans cette table nous avons enregistré le mot choisi par l'utilisateur et son nombre de fois il a été utilisé dans la recherche.

4.2 les classes :

Dans le package de notre application nous avons deux classes :

- **Class Principale :** qui loge la classe « main » de notre projet qui exécute notre interface d'application.

- **Classe Fenêtre :** elle affiche l'interface de notre projet elle fait appel à toutes les procédures et les méthodes du système et dans l'exemple suivant un partie de code sur une telle opération que ce fait dans cette interface.

- **Classe visualisation :** cette classe contient une interface de tableau qui permet à l'utilisateur de visualiser un tableau contenant deux colonnes la première sauvegarde le mot introduit et la deuxième le nombre de fois il a été introduit.

4.3 Bibliothèque :

Nous avons utilisé deux bibliothèques dans notre projet sont :

4.3.1 Bibliothèque UCanAccess :

C'est une implémentation de pilote Java JDBC open-source qui permet aux développeurs Java et aux programmes clients JDBC (par exemple, DBeaver, NetBeans, SQLeo, base OpenOffice, base LibreOffice, Squirrel SQL) de lire / écrire des bases de données Microsoft Access [web1].

Nous avons utilisé la version UCanAccess 4.0.3 et ci-après toutes les étapes suivies pour l'intégrer au projet :

Premièrement, on cible notre projet en l'ouvrant sur un IDE (éditeur de langage Netbeans dans notre cas) ce projet a sa bibliothèque originale propre a lui (**Libraries**), le but

ainsi et de faire immigrer les JAR (des fichiers exécutables java) et les importer sur notre application pour une éventuelle future utilisation (démonstration en ***Figure2.2*** et ***Figure2.2***).

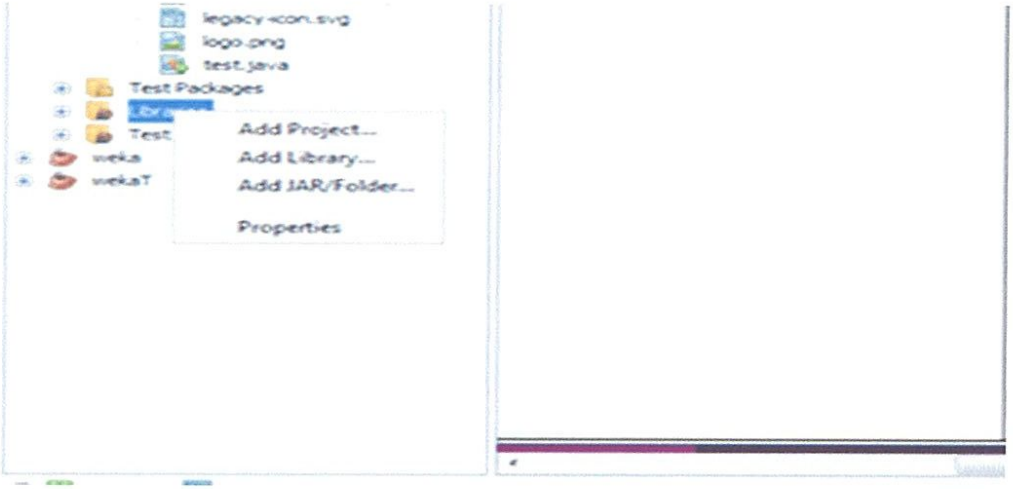


Figure2.2 : *l'ajout de la bibliothèque UCanAccess.*

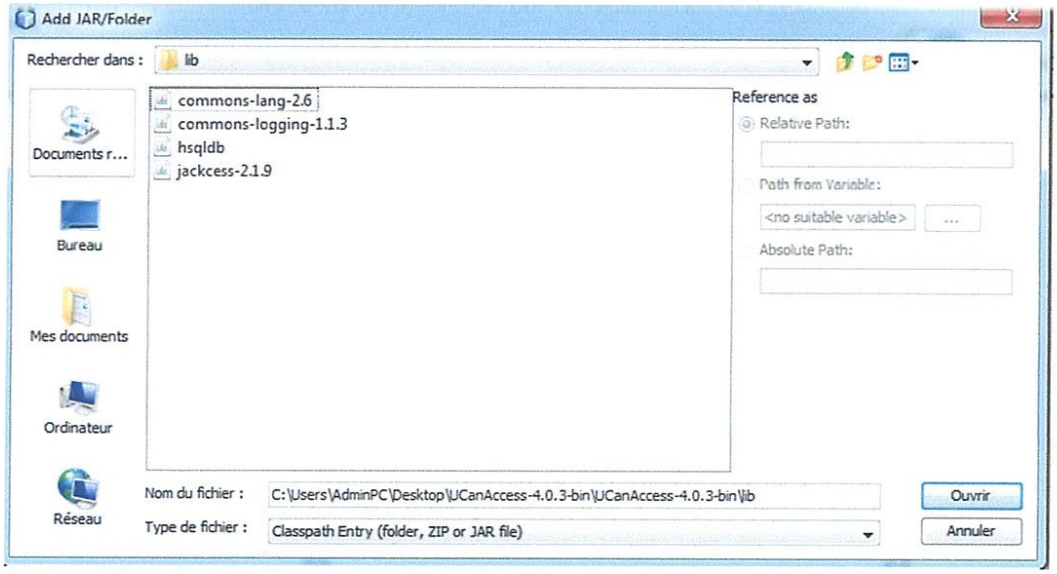


Figure2.3 : *les JAR de bibliothèque UCanAccess.*

4.3.2 Bibliothèque Weka :

Weka est un ensemble d'algorithmes d'apprentissage automatique pour les tâches d'exploration de données. Les algorithmes peuvent être appliqués directement à un ensemble de données ou appelés à partir de votre propre code Java.Weka contient des outils pour le prétraitement des données, la classification, la régression, le clustering, les règles d'association et la visualisation.

Il est également bien adapté au développement de nouveaux schémas d'apprentissage machine [web5].

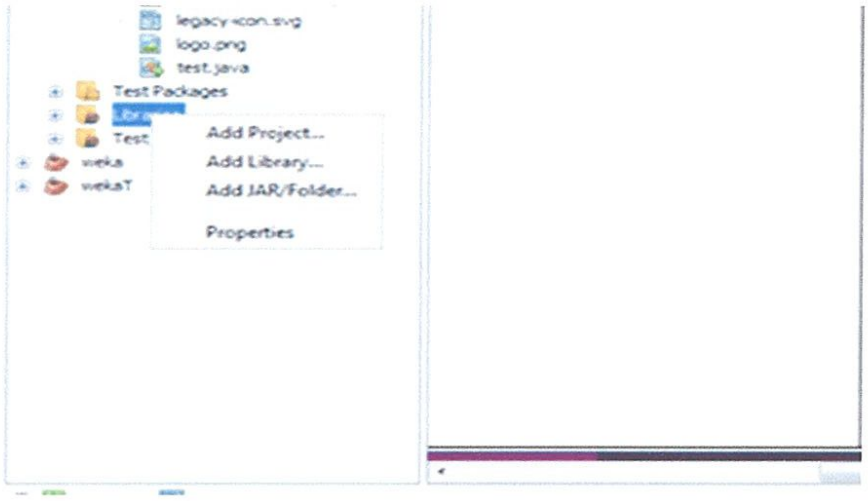


Figure2.4 : *l'ajout de la bibliothèque Weka.*

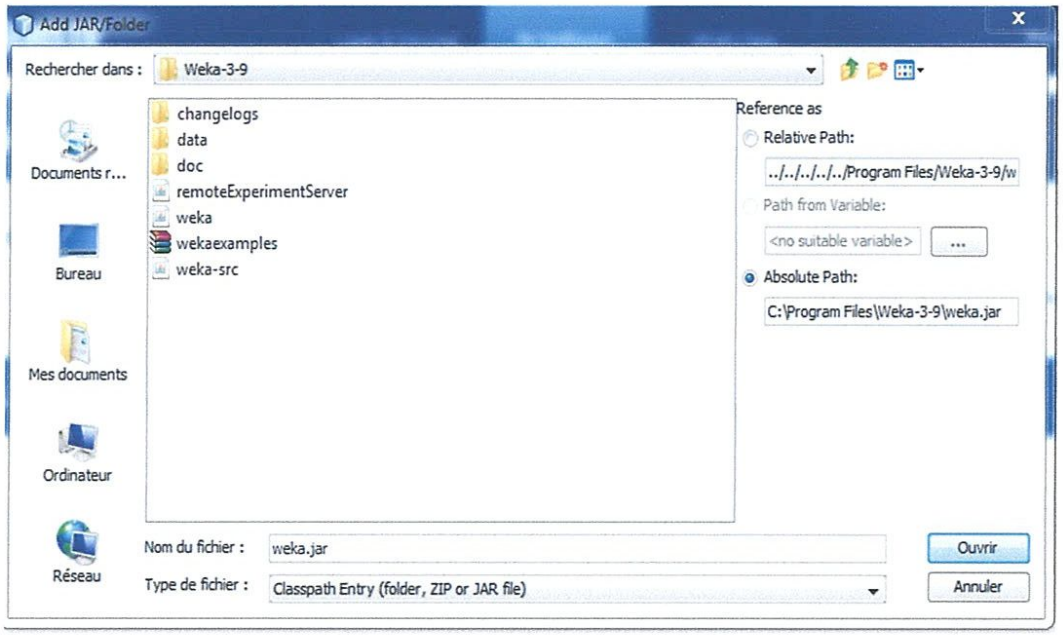


Figure2.5: *le JAR de bibliothèque Weka.*

Les algorithmes d'apprentissage dans Weka sont dérivés de la classe abstraite:

- weka.classifiers.Classifier

Algorithme SMO :

SMO (Sequential minimal Optimization) définit la méthode des SVM.

SMO est un algorithme itératif pour résoudre le problème d'optimisation. SMO divise ce problème en une série de sous-problèmes les plus petits possibles, qui sont ensuite résolus analytiquement. En raison de la contrainte d'égalité linéaire impliquant les multiplicateurs de Lagrange, le plus petit problème possible implique deux de ces multiplicateurs [web3].

4.4 La pseudo-réinjection de pertinence :

A partir de la requête initiale, le système a identifié la classe d'appartenance et à partir de cette classe les requêtes les plus proches de la requête initiale sont sélectionnées. Le critère de sélection de base est la cooccurrence telle que les mots de la classe qui apparaissent le plus avec les mots de la requête initiale seront choisis pour l'expansion et seront affichées dans un ordre de cooccurrence décroissant. L'utilisateur donne ensuite ces jugements de significativité à propos les mots proposés pour l'expansion de sa requête. Les tests ont été faits et leur résultat sera présenté dans le dernier chapitre.

5. Conclusion :

Dans ce chapitre nous avons présenté l'architecture de notre système ainsi que les étapes de sa conception à savoir : la phase de collecte de données, la phase de prétraitement, la phase d'apprentissage la phase de classification, la phase d'expansion et la phase de test dont leur résultat seront présenté en chapitre implémentation.

Chapitre 3 :

Implémentation

1. Introduction

Dans ce chapitre nous exposons les phases d'implémentation du système proposé permettant de proposer des expansions pertinentes des requêtes des utilisateurs en faisant appel a un algorithme de classification fiable SVM.

2. Les outils de développement

2.1 Plateforme Matérielle :

L'implémentation de l'application est réalisée sur un ordinateur portable ayant les caractéristiques suivantes :

- Machine : HP
- Processeur : Intel(R) Core (TM) i3-2348M CPU
- Fréquence : 2.30 GHz
- RAM : 4.00 Go
- Carte graphique : Intel(R)
- Système d'exploitation : Microsoft Windows 7 édition Intégrale.

2.2 Plateforme logicielle :

- Le Système d'exploitation choisi pour la réalisation de notre application est le Microsoft **Windows 7 édition Intégrale**. Ce choix repose sur le fait que ce système possède tout la puissance la stabilité et surtout la cohérence avec le java, ce système offre également la possibilité de développer rapidement des applications qui nécessitent l'interaction avec une base de donnée **Access** ce dernier fait partie du pack « Microsoft office » destiné à loger principalement sur des plateformes Windows.

- Le langage de programmation choisi pour développé notre application était le langage «**Java**» qui est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, deux employés de la société Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers

plateformes et Frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java [web8].

- L'environnement du développement choisi pour réaliser notre application est « NetBeans », nous avons utilisé l'environnement **NetBeans** qui est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de *greffons*. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

NetBeans constitue par ailleurs une plate-forme qui permet le développement d'application spécifique (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme [web9].

La version qui nous utilisons NetBeans IDE 8.2.

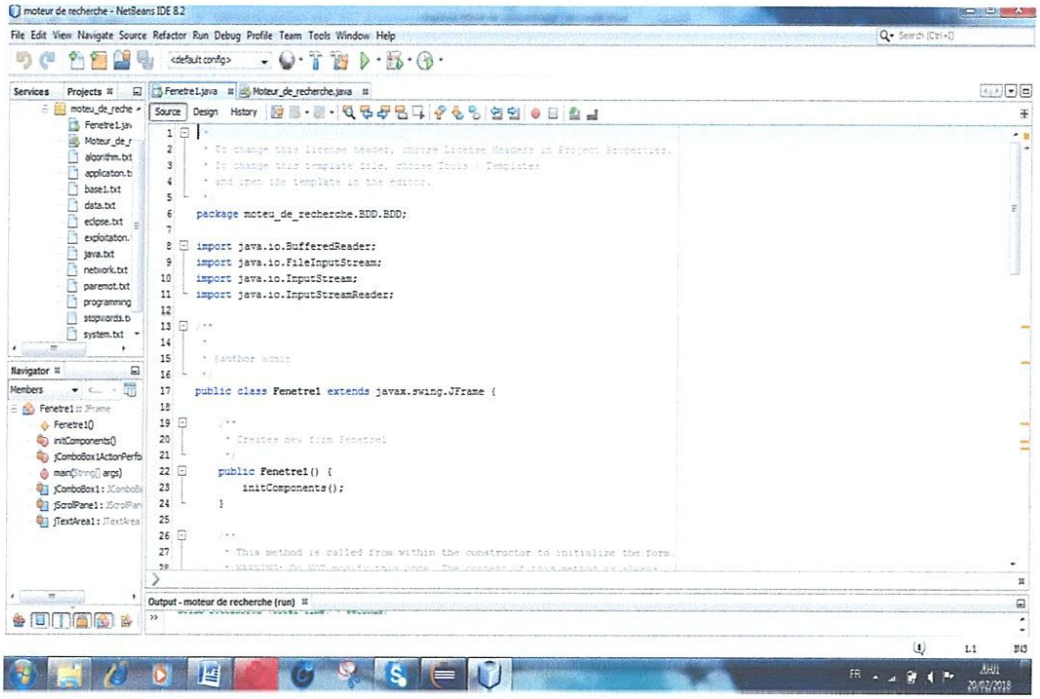


Figure 3.1: *L'environnement NetBeans IDE 8.2.*

2.2.1 Weka :

Weka (Waikato Environment for Knowledge Analysis) est un ensemble d'outils permettant de manipuler et d'analyser des fichiers de données, implémentant la plupart des algorithmes SVM, dont les arbres de décision et les réseaux de neurones [web10].

Il est écrit en java, et s'appuie sur le livre Data Mining , practical machine learning tools and techniques with Java implementations:

Il se compose principalement:

- De classes Java permettant de charger et de manipuler les données.
- De classes pour les principaux algorithmes de classification supervisée ou non supervisée.
- D'outils de sélection d'attributs, de statistiques sur ces attributs.
- De classes permettant de visualiser les résultats.

On peut l'utiliser a` trois niveaux :

- Via l'interface graphique, pour charger un fichier de données, lui appliquer un algorithme, vérifier son efficacité.
- Invoquer un algorithme sur la ligne de commande.
- Utiliser les classes définies dans ses propres programmes pour créer d'autres méthodes, implémenter d'autres algorithmes, comparer ou combiner plusieurs méthodes [web10].

Figure 3.2 : *Plateforme Weka.*

3. Description des fichiers utilisés

Notre base contient 10 fichiers en langue anglaise. Chaque fichier comporte une requête de recherche menée par un seul mot sur Google et la première page de résultat est

récupérée. On collecte ces résultats et on construit un document qui contient des titres, des liens et une description de chaque résultat.

Le tableau suivant montre une description de notre base:

N°	fichiers	21	System meaning
1	Algorithm	22	System definition
2	Application	23	Computer system
3	Data	24	Programmer developer
4	Network	25	Exploitation definition
5	Exploitation	26	Network networking
6	System	27	Exploitation function
7	Programming	28	Exploitation act
8	Base	29	Exploitation natural
9	Java	30	Data memory
10	Eclipse	31	Data analysis
11	Algorithm procedure	32	Raw data
12	Algorithm computer	33	Base added
13	Algorithm definition	34	Base synonyms
14	Application program	35	Base antonyms
15	Application online	36	Oracle technology
16	Application process	37	Data computing
17	Java programs	38	Network ultimate
18	Technology network	39	Technical information
19	Computer programming	40	Base opposite
20	Java Development	41	Eclipse downloads

Tableau 3.1 : la base des fichiers du système.

4. Description du système

Dans cette section nous présenterons les différentes interfaces de notre système prises par des captures d'écran.

Au lancement de l'application, la fenêtre suivante s'affiche:

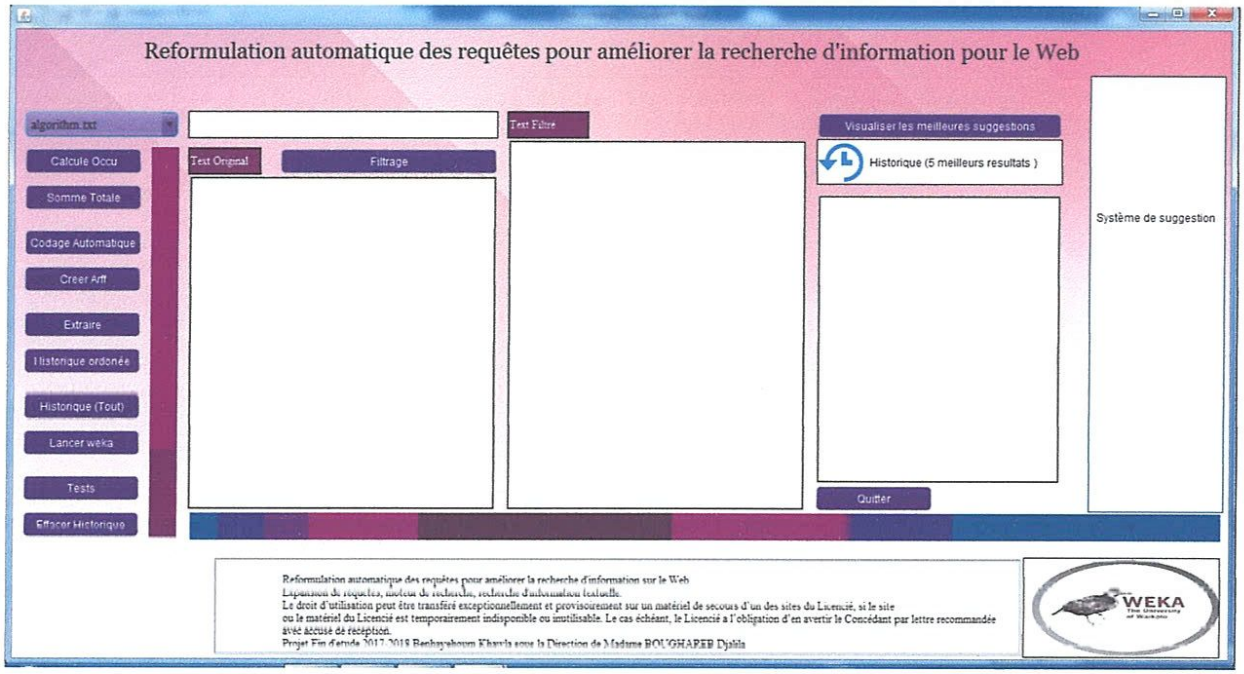


Figure 3.3 : *Interface notre du système.*

➢ **ComboBox :** permet de choisir un fichier parmi plusieurs fichiers existant. Le résultat est affiché dans l'emplacement **Text Original.**

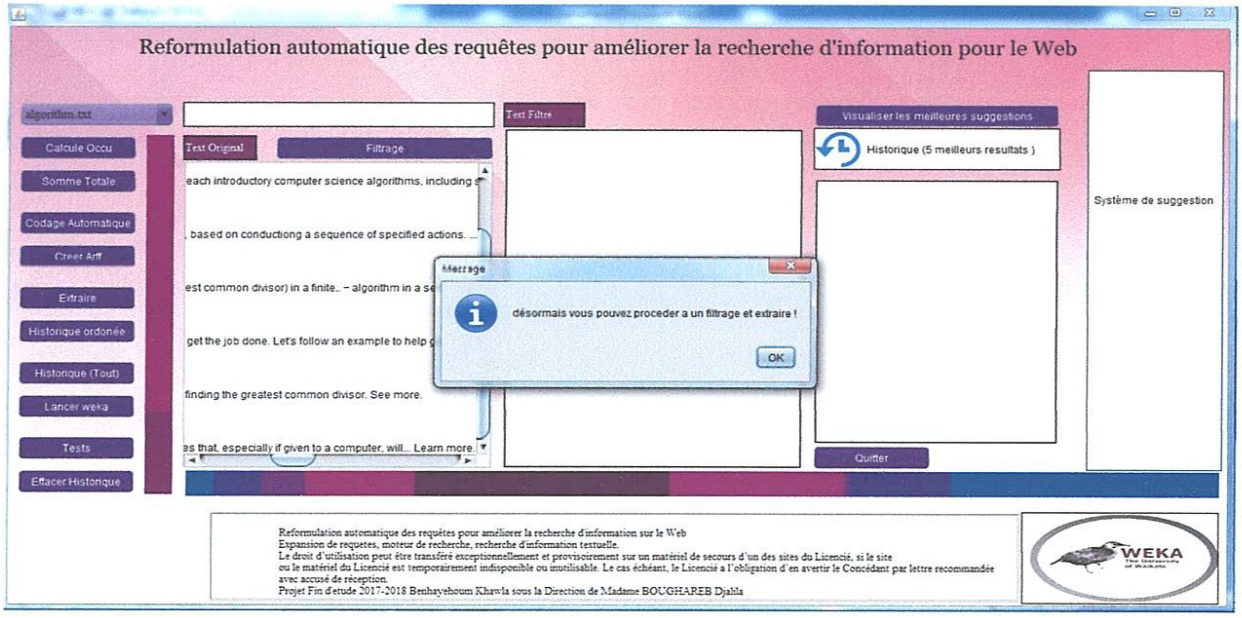


Figure 3.4 *: Le résultat de choix de fichier.*

➢ **Filtrage :** sa fonction est d'éliminer les mots vides (stop word qui existe dans le fichier) et affiche le résultat dans l'espace **Text Filtré**.

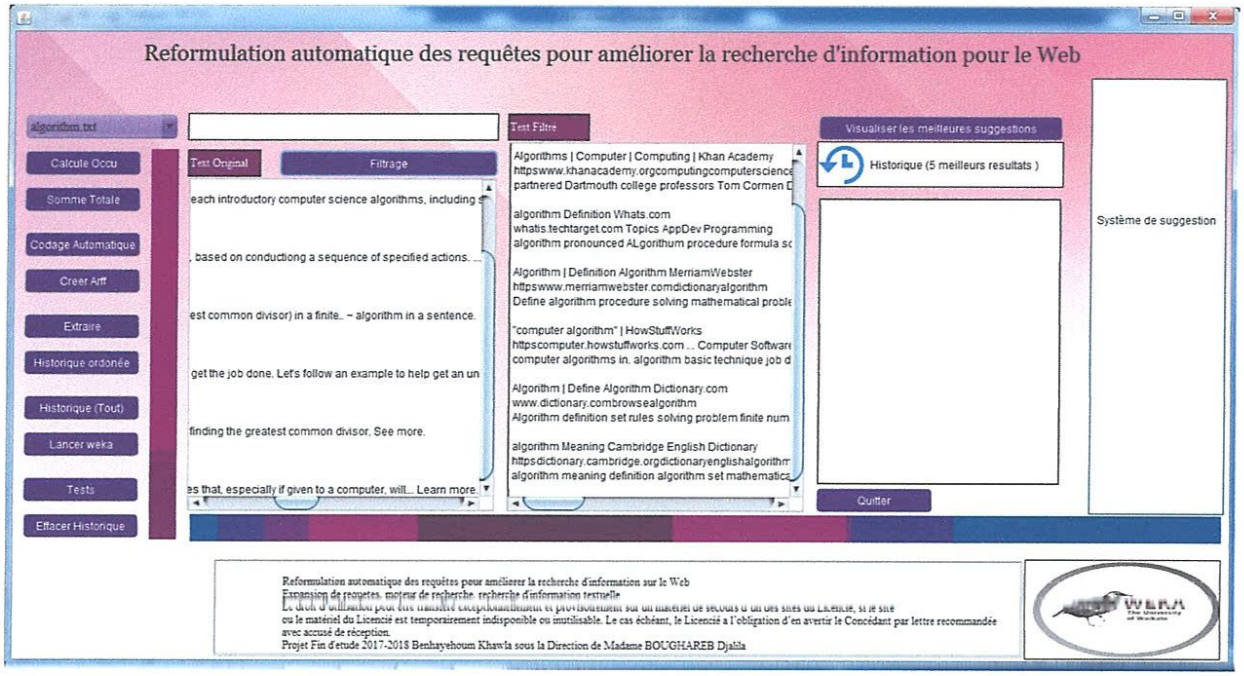


Figure 3.5 : *Fichier choisi après le filtrage.*

➢ **Calcule d'occurrence :** calculer le nombre d'occurrence d'un mot dans chaque phrase du fichier (le nombre de fois le mot apparait dans le fichier).

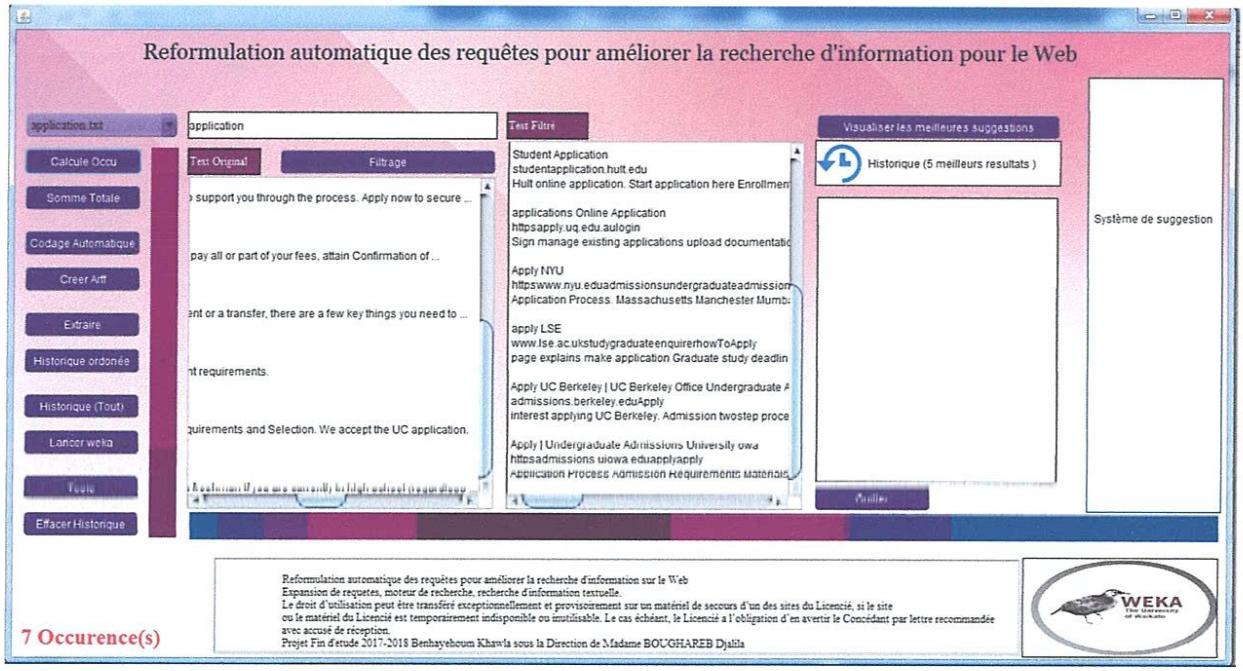


Figure 3.6 : *Nombre d'occurrence d'un mot dans un fichier.*

➢ **Somme totale :** calcule le nombre d'occurrence d'un mot dans tous les fichiers de la collection.

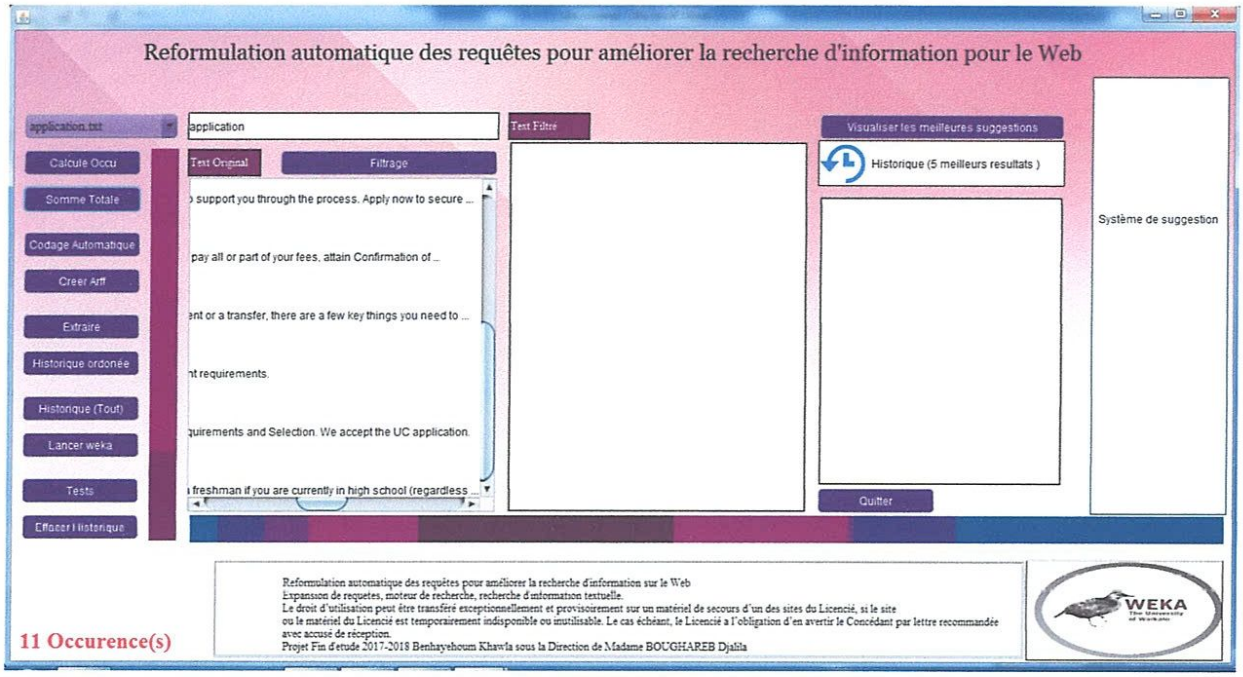
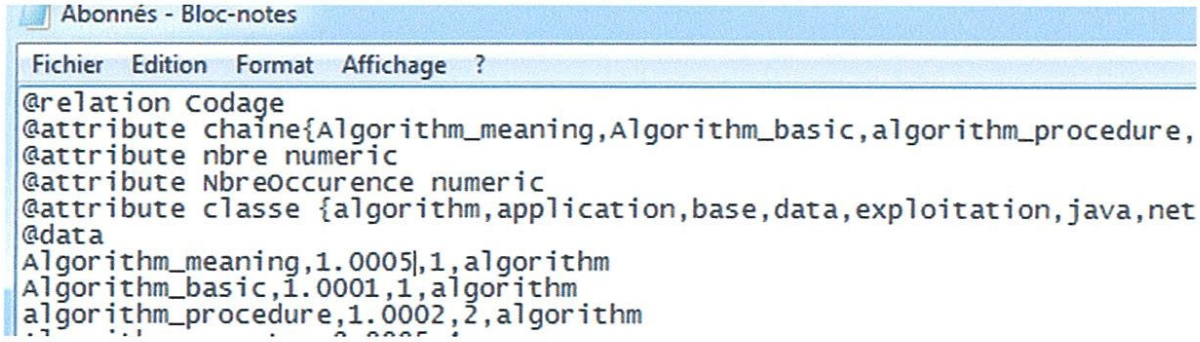


Figure 3.7: *Nombre d'occurrences dans la collection.*

➢ **Codage automatique :** elle permet de coder tout les paires de mots clés extraits de la collection et de les enregistrer dans la base en mentionnant la paire, son code, son nombre d'occurrence et sa classe d'appartenance.

	Algorithm_meaning	1.0005	1	algorithm
	Algorithm_basic	1.0001	1	algorithm
	algorithm_procedure	1.0002	2	algorithm

➢ **Créer arff :** créer le fichier au format (*.arff) pour faire leur appel dans le weka avec les données enregistrées à l'étape de codage automatique. Ce fichier sera utilisé pour faire un apprentissage automatique et créer un système de classification automatique des mots et des paires de mots.



➢ **Extraire :** extrait à partir la base de données le mot et leur cooccurrence dans un ordre décroissant.

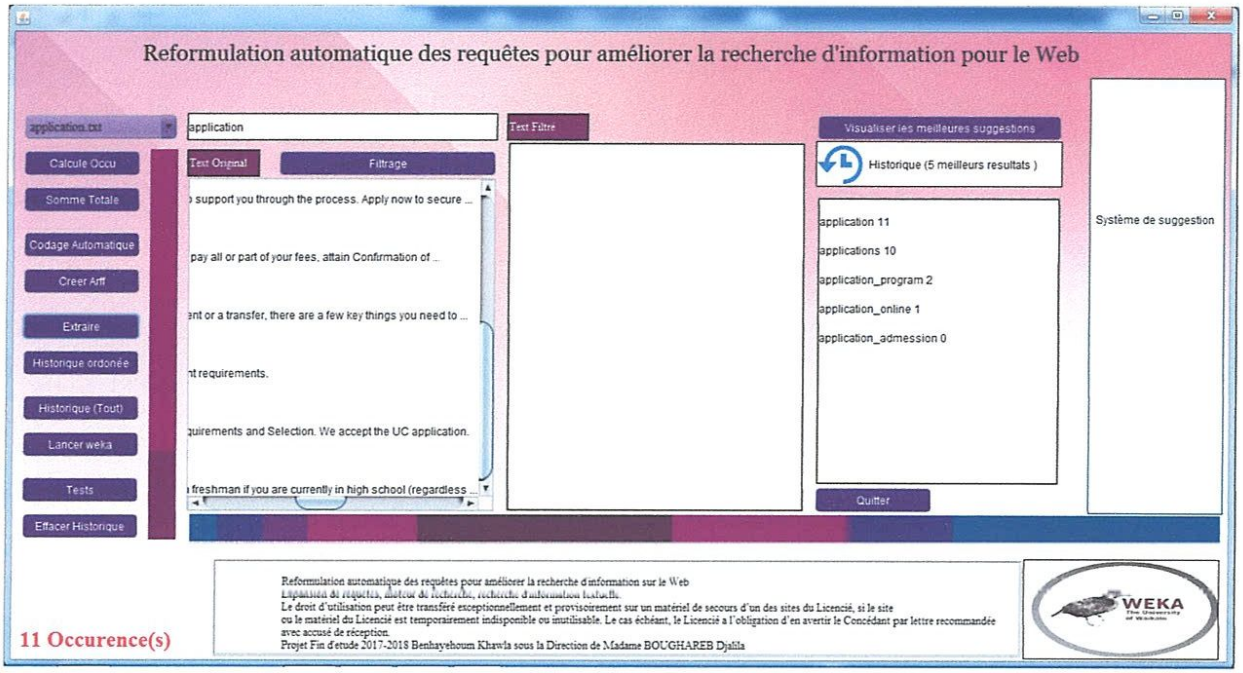


Figure 3.8: *Extrait le nombre d'occurrence dans ordre décroissant.*

➢ **Historique Ordonnée** va affichée tout les mots qui existent dans la base de données dans un ordre décroissant.

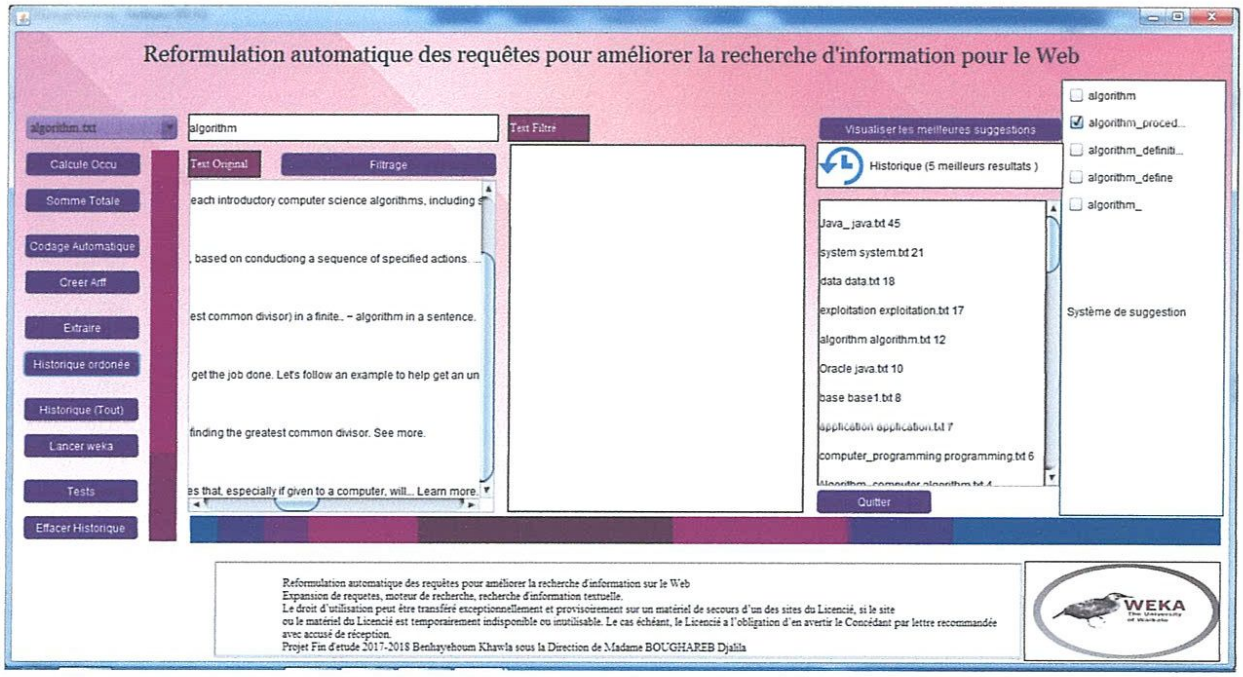


Figure 3.9: *L'affichage de l'historique.*

➢ **Lancer WEKA :** pour faire appel à weka. Et afficher le résultat d'apprentissage supervisé fait âpres avoir attribué manuellement les mots et les paires de la base d'apprentissages à des classes préalablement créer.

Nous avons 11 classes : classe algorithme, classe application, classe java, classe data, classe base, classe network, classe exploitation, classe system, classe eclipse, classe programming, classe other.

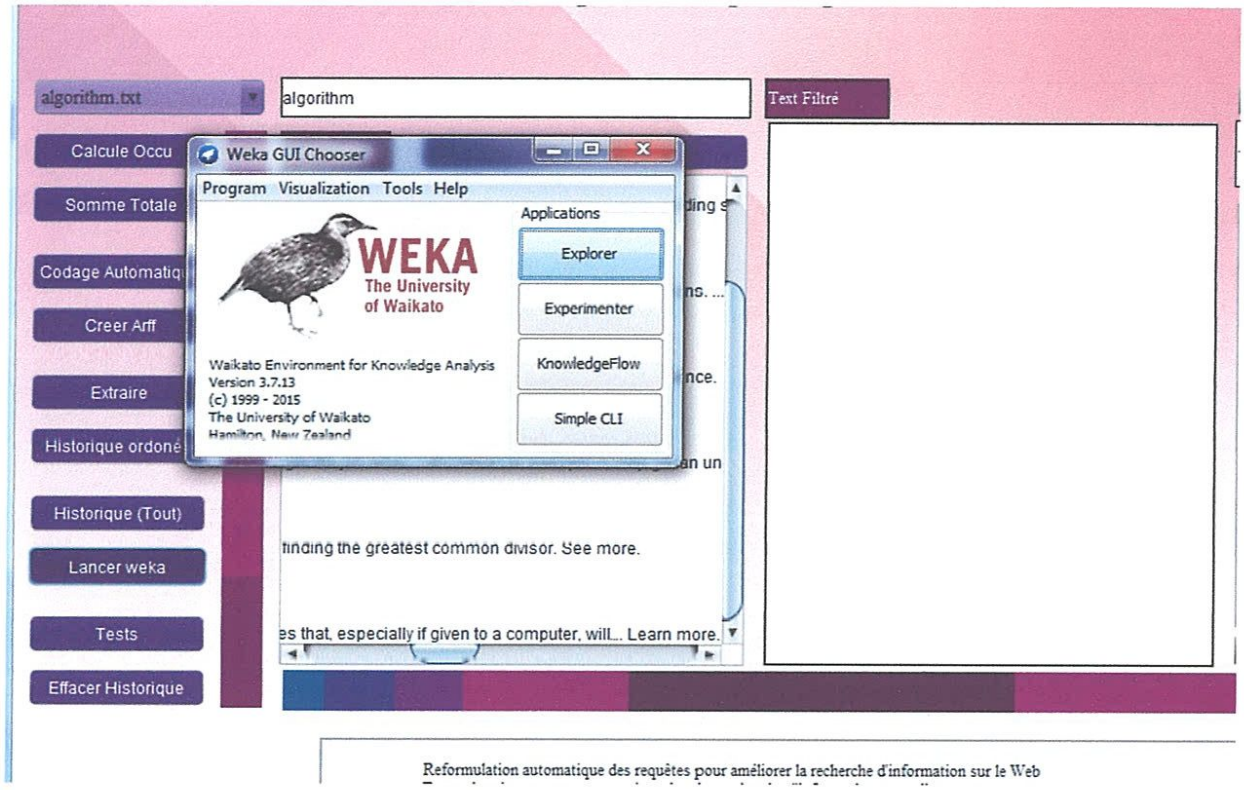


Figure3.10:*Fenêtre lancement de Weka.*

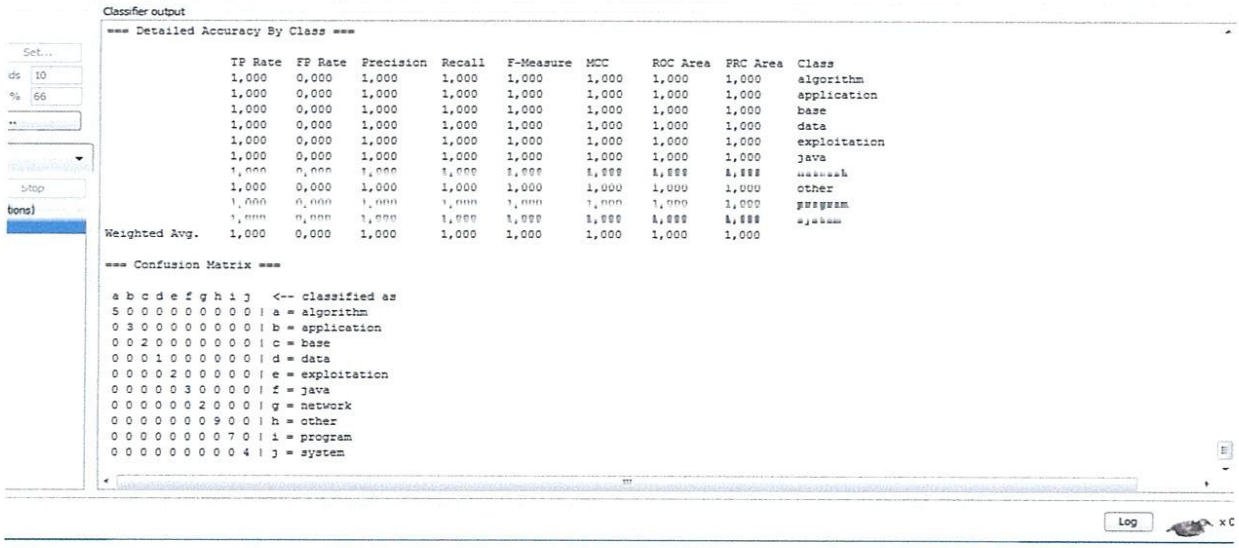


Figure3.11:*Résultats de lancement d'algorithme SMO.*

➢ **Tests :** permet à l'utilisateur d'afficher les choix dans ordre décroissant pour faire la sélection des mots clés pertinents parmi la liste des mots clés suggéré par le système. L'utilisateur après avoir tapé la requête le système lui suggère des mots clés susceptibles de lui aider à étendre sa requête initiale. Ensuite l'utilisateur peut évaluer le résultat du système par cocher les plus proches à son besoin en information.

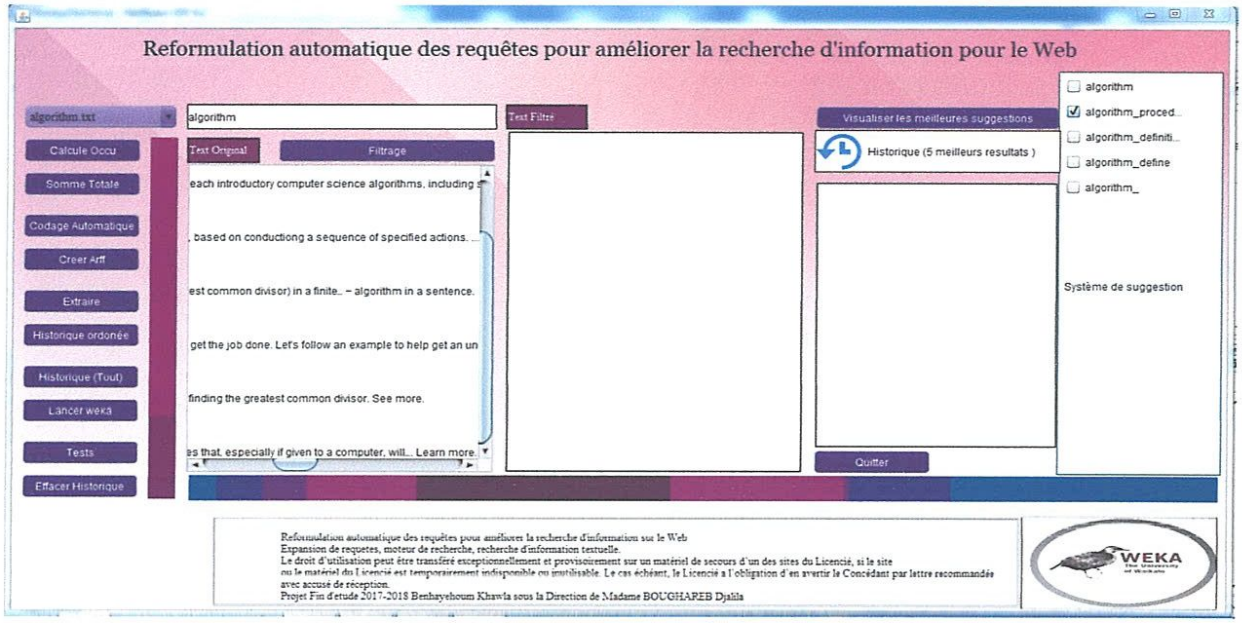
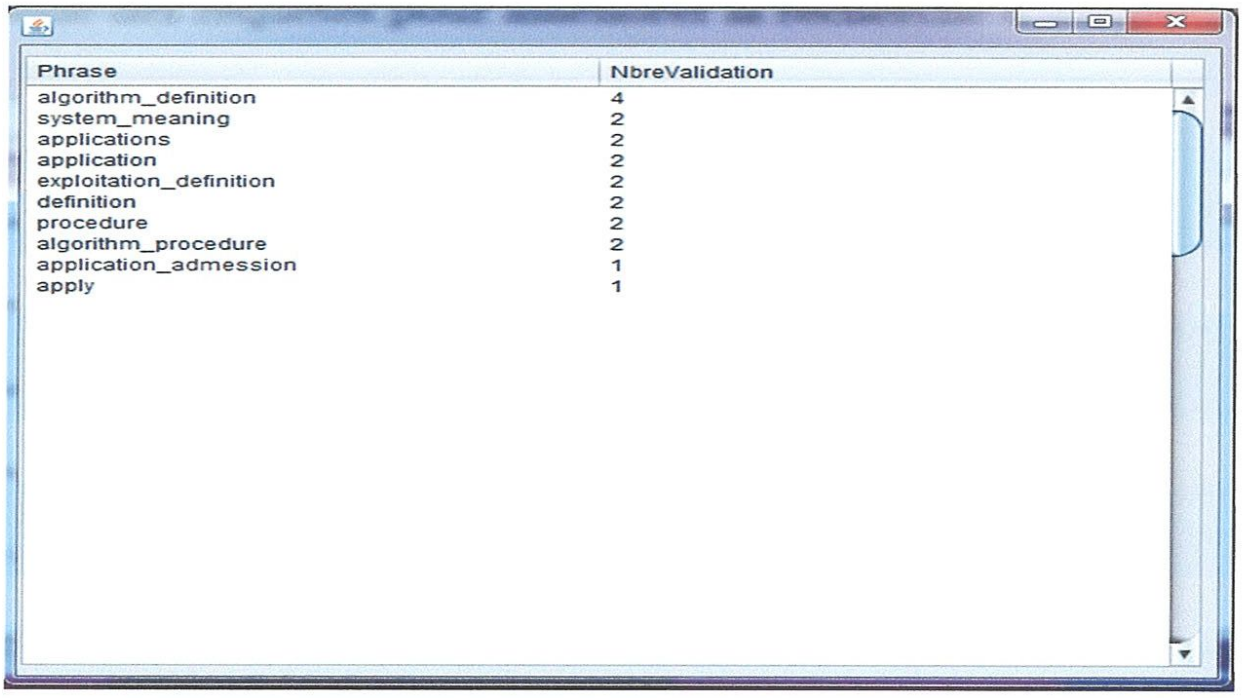


Figure3.12 : *Résultats de tests.*

➢ **Visualiser les meilleures suggestions :** il affiche dans un tableau le résultat d'évaluation des suggestions par l'utilisateur.



Phrase	NbreValidation
algorithm_definition	4
system_meaning	2
applications	2
application	2
exploitation_definition	2
definition	2
procedure	2
algorithm_procedure	2
application_admession	1
apply	1

Figure3.1 3 : *visualisation des tests.*

➢ **Tests classe :** cette action est affichée les mots qui appartient au même classe de le mot requête initiale.

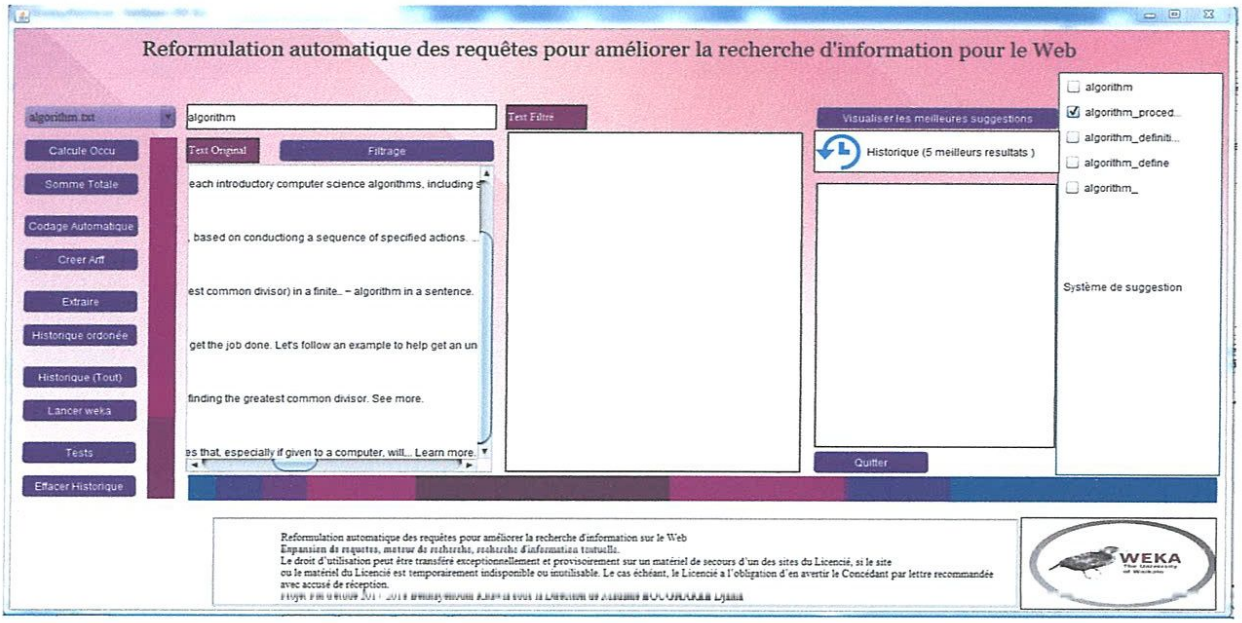


Figure3.14 : *Résultats de tests classe.*

5. Expérimentation et Résultats

Cette section présente les résultats de l'utilisation de l'expansion de requête et la classification.

- **Evaluation :**

Le système associe pour chaque nouvelle requête son propre classe et présente les résultats en forme de liste de choix.

Les choix suggérés par le système se basent sur une heuristique qui consiste à la sélection des top-k des mots extraits de documents appartenant à la même classe d'appartenance.

Afin de savoir si le travail que nous avons réalisé répond aux besoins de l'utilisateur. nous avons mené des tests où on a engagé cinq utilisateurs. On affiche les requetes à etendre et les résultats d'evaluations comme suit :

- Pour le choix sur 10 requetes reformulées :

Requêtes	Tests	Rappel @10	Tests classe	Rappel @10

algorithm	5/10	0.5	7/10	0.7
programming	4/10	0.4	3/10	0.3
definition	2/10	0.2	2/10	0.2
language	1/10	0.1	4/10	0.4
network	4/10	0.4	6/10	0.6
information	3/10	0.3	3/10	0.3
télécommunications	2/10	0 .2	3/10	0.3
java	6/10	0.6	8/10	0.8
apply	5/10	0.5	6/10	0.6
meaning	2/10	0.2	4/10	0.4
online	4/10	0 .4	6/10	0.6
installer	3/10	0.3	4/10	0.4
Technology	3/10	0 .3	5/10	0.5
version	2/10	0.2	4/10	0.4
Developer	6/10	0.6	6/10	0.6
exploiting	7/10	0.7	6/10	0.6
systematic	8/10	0.8	7/10	0.7
data	7/10	0.7	6/10	0.6
oracle	4/10	0.4	4/10	0.4
system	8/10	0.8	7/10	0.7

Tableau 3.2 : *Evaluation de pertinence des requêtes.*

Dans la section tests, tout d'abord dans le Tests classe on travaille sur les classes qui existent. On entre une requête avec un seul mot après il nous donne des suggestions à partir la classe qui le mot il appartient.

Et dans autre part de Tests (par cooccurrence), dans cette partie ne travaille pas avec les classes. On travaille sur le nombre d'occurrence il affiche des suggestions sans classification.

Rappel moyen Tests = 0.43

Rappel moyen Tests classe = 0.505

- Pour le choix sur 10 requetes reformulé :

Requetes	Tests	Rappel @5	Tests classe	Rappel @5
algorithm	4/5	0.8	4/5	0.8
programming	2/5	0.4	3/5	0.6
definition	3/5	0.6	2/5	0.4
language	1/5	0.2	3/5	0.6
network	4/5	0.8	3/5	0.6
information	3/5	0.6	3/5	0.6
télécommunications	2/5	0 .4	3/5	0.6
java	5/5	1	5/5	1
applying	3/5	0.6	2/5	0.4
procedure	2/5	0.4	4/5	0.8
Working	2/5	0 .4	3/5	0.6
installer	3/5	0.6	4/5	0.8
Technology	3/5	0 .6	2/5	0.4
topolgy	2/5	0.4	4/5	0.8
Developer	1/5	0.2	3/5	0.6
exploiting	2/5	0.4	3/5	0.6
systcmatic	2/5	0.4	1/5	0.2
environment	2/5	0.4	1/5	0.2
oracle	4/5	0.8	4/5	0.8
system	3/5	0.6	2/5	0.4

Tableau 3.3 : *Evaluation de pertinence des requêtes.*

Rappel moyen Tests = 0.53

Rappel moyen Tests classe = 0.59

- **Discussion des résultats :**

Nous avons évalué notre classifieur en se basant sur l'évaluation d'une collection de 41 resultats obtenus depuis google et dans la majorité des cas les résultats sont satisfaisants. Puis, dans la requete **« Data »** le rappel dans la requete data avec le test de classe 0,7 est très important et dans le rappel test est sont modestes et égale à 0.6 parce que le nombre de documents traités appartenant à cette classe décent.

A ce stade nous pouvons dire que la classification supervisée et à base des resultats de recherches presentés par google utilisés comme source de la pseudo-reinjection de pertinenece nous a permis d'améliorer les requetes à mots uniques dans lebut de se rapprocher du besoin informationnel de l'utilisateur.

6. Conclusion

D'après les tests, on pouvt voir que l'expansion de requete avec la classification (l'agorithme d'apprentissage SVM) et le pseudo réinjection de pertinence a donne des meilleurs résultats que la methode bas é seulement sur la coocurence de mots.

Conclusion générale

Conclusion générale

Le thème que nous avons abordé dans ce mémoire s'intéresse à l'expansion automatique des requêtes pour améliorer la recherche d'information sur le web. L'idée principale de notre approche qui constitue une originalité est d'utiliser les résultats de recherche fournis par le moteur de recherche Google comme une source pour la pseudo-réinjection de pertinence.

Une telle démarche est naturelle car les utilisateurs qui explorent le web recourent à plusieurs moyens de sélection des documents intéressent.

La collecte d'information s'effectue dans un cycle décrit par le processus qui comprend la recherche de documents pertinents l'extraction où nous nous sommes particulièrement intéressés à l'analyse des requêtes, de point de vue de leur formulation.

Nous sommes focalisés sur la proposition de modèles d'expansion de requêtes avec le pseudo réinjection de pertinence. Ainsi, notre travail se penche sur la réalisation de ceci, ainsi à l'aide d'une classification préalablement obtenue par une méthode de classification quelconque (dans notre cas l'Algorithme de SVM) on pourrait désormais mettre l'accent sur l'une des requêtes de voisinages.

Le principal objectif de notre système est de proposer de nouveaux mots clés à la requête initiale afin de l'enrichir et de rapprocher du besoin informationnel de l'utilisateur.

A la fin de ce travail on se voit obligé de citer quelques problèmes rencontrés avec la méthode de pseudo réinjection de pertinence :

- Difficulté à reformuler la requête car il est possible de s'éloigner de la signification désirée.
- L'existence de mots qui se chevauchent entre les documents (résultats de recherche préalablement obtenus) rend leur classification difficile et donne donc des résultats erronés lors de l'expansion.

Pour cela, nous avons pensé à enrichir la collection de données afin d'élargir l'application de l'approche proposée et pour étendre le test sur un ensemble d'intérêts plus large.

Bibliographie

Bibliographie

[ADR99] : Adriani, M. et Rijsbergen, C. J. V. (1999). Term similarity-based query expansion for crosslanguage information retrieval. In In Proceedings of the third European Conference on Research and Advanced Technology for Digital Libraries (ECDL '99, pages 311–322.

[ARO17] : Arora P., Foster J., Jones G.J.F. (2017) Query Expansion for Sentence Retrieval Using Pseudo Relevance Feedback and Word Embedding. In: Jones G. et al. (eds) Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2017. Lecture Notes in Computer Science, vol 10456. Springer, Cham.

[BAE99] : Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison Wesley.Bai, J., Nie, J.-Y., and Cao, G. 2006. Context- dependent term relations for information retrieval.In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Sydney, Australia, pp.551– 559.

[BAI05] : Bai, J., Song, D., Bruza, P., Nie, J.-Y., and Cao, G. (2005). Query expansion using term relationships in language models for information retrieval. In Proceedings of the 14th ACM international conference on Information and knowledge management. ACM Press, Bremen, Germany, pp. 688–695.

[BAI06] : Bai. J., Nie, J.-Y., and Cao, G. (2006). Context-dependent term relations for information retrieval. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Sydney, Australia, pp.551–559.

[BAZIZ03] : Baziz, M., Aussenac-Gilles, N. et Boughanem, M. (2003). Désambiguisation et Expansion de Requêtes dans un SRI,Etude de l'apport des liens sémantiques. Revue des Sciences et Technologies de l'Information (RSTI) série ISI, 8(4/2003):113–136.

[BEE00] : Beeferman, D. and Berger, A. (2000). Agglomerative clustering of a search engine query log. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, Boston, MA, USA, pp. 407–416.

[BEL00] : Belkin, N.J. (2000) Prospects for information "selection". Presentation for UCAO, March 2000.

[BHO07] : Bhogal, J., Macfarlane, A., and Smith, P. (2007). A review of ontology based query expansion. Information Processing and Management 43(4), pp.866–886.

[BIA09] : Biancalana, C., Lapolla, A., & Micarelli, A. (2009). Personalized Web Search Using Correlation Matrix for Query Expansion, in J. Cordeiro et al. (Eds.): WEBIST 2008, LNBIP 18, pp. 186–198.

[BIL03] : Billerbeck, B., Scholer, F., Williams, H. E., and Zobel, J. (2003). Query expansion using associated queries. In Proceedings of the 12th ACM international conference on Information and knowledge management. ACM Press, New Orleans, Louisiana, USA, pp.2–9.

[Boughareb 2014] : Boughareb,D. Thèse Doctorat Recherche d'information multicritères chapitre 4,un aperçu des déférentes techniques de reformulation de requête , pp.44-52.

[BOU13b] : Boughareb, D., and Farah, N. (2013b). A Query Expansion Approach Using the Context of the Search, Advances in Intelligent and Soft Computing (Springer),

Book chapter. In the 4th International Symposium on Ambient Intelligence, Salamanca-Spain ISami'13, Volume 219, DOI. 10.1007/978-3-319-00566-9_8. ISBN. 978-3-319-00566-9, pp. 57-63.

[BOU99] : Boughanem, M. Chrisment, C. and Soule-Dupuy. C. (1999). Query modification based on relevance backpropagation in adhoc environment. Information Processing and Management, 35. pp. 121-139.

[BUCK92]: Buckley, C., Salton, G. et Allan, J. (1992). Automatic retrieval with locality information using smart. In Harman, D. K., éditeur : TREC, volume Special Publication 500-207, pages 59–72. National Institute of Standards and Technology (NIST).

[CAR01] : Carpineto, C., De Mori, R., Romano, G., and Bigi, B. (2001). An Information Theoretic Approach to Automatic Query Expansion. ACM Transactions on Information Systems (TOIS) 19(1):1–27.

[CARP02]: Carpineto, C., Romano, G. et Giannini, V. (2002). Improving retrieval feedback with multiple term-ranking function combination. ACM Trans. Inf. Syst., 20(3):259–290.

[CARP12]: Carpineto, C. et Romano, G. (2012). A survey of automatic query expansion in information retrieval. ACM Comput. Surv., 44(1):1 :1–1 :50.

[CHI07] : Chirita, P.-A., Firan, C. S., and Nejdl, W. (2007). Personalized query expansion for the web. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Amsterdam, The Netherlands, pp. 7–14.

[CUI03] : Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2003). Query expansion by mining user logs. IEEE Transactions on Knowledge and Data Engineering 15(4):829–839.

[CURé13]: Curé, O., Maurer, H., Shah, N. et LePendu, P. (2013). Refining health outcomes of interest using formal concept analysis and semantic query expansion. In Proceedings of the 7th International Workshop on Data and Text Mining in Biomedical Informatics, DTMBIO '13, pages 5–6, New York, NY, USA. ACM.

[DIA06] : Diaz, F. and Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Seattle, Washington, USA, pp.154– 161.

[DUAN16]: Duan, L., Dong, S., Cui, S. et Ma, W. (2016). Proceedings of ELM-2015 Volume 1 : Theory, Algorithms and Applications (I), chapitre Extreme Learning Machine with Gaussian Kernel Based Relevance Feedback Scheme for Image Retrieval, pages 397–408. Springer International Publishing, Cham.

[EFTH96]: Efthimiadis, E. N. (1996). Query expansion. Annual review of information science and technology, 31:121–187.

[EYA16]: Eya Znaidi. (30/06/2016). Contribution à l'analyse et l'évaluation des requêtes expertes : cas du domaine médical. Chapitre1 : Recherche d'information : Concepts et modèles pp. 29-33.

[FERN16] : Fernandez-Beltran, R. et Pla, F. (2016). Latent topics-based relevance feedback for video retrieval. Pattern Recognition, 51:72 – 84.

Bibliographie

[FON05] : Fonseca, B. M., Golgher, P. B., Possas, B.,Ribeiro-Neto, B. A., and Ziviani, N. (2005). Concept-based interactive query expansion. In CIKM, pp. 696-703.

[FUR87]: Furnas, G. W., Landauer, T. K., Gomez, L. M. et Dumais, S. T. (1987). The vocabulary problem in human-system communication. Commun. ACM, 30(11):964–971.

[HAMM13] : Hammache, A., Boughanem, M. et Ahmed-Ouamer, R. (2013). Pseudo-réinjection de pertinence basée sur un modèle de langue mixte combinant les termes simples et composés. In CORIA 2013- Conférence en Recherche d'Infomations et Applications - 10th French Information Retrieval Conference, Neuchâtel, Suisse, April 3-5, 2013., pages 175–190.

[HAO17] : Haolin W., Qingpeng Z., Jiahu Y. (Reçu le 1er mars 2017, accepté le 14 avril 2017, date de publication 26 avril 2017, date de la version actuelle 7 juin 2017) Semantically Enhanced Medical Information Retrieval System: A Tensor Factorization Based Approach.

[HAR92] : Harman. D. (1992). Relevance feedback revisited. In 15th Annual International ACM SIGIR Conference on Research and development in Information Retrieval, pp. 1-10.

[HAZI15]: Hazimeh, H. et Zhai, C. (2015). Axiomatic analysis of smoothing methods in language models for pseudo-relevance feedback. In Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR '15, pages 141–150, New York, NY, USA. ACM.

[Huang09]: Huang, J. et Efthimiadis, E. N. (2009). Analyzing and evaluating query reformulation strategies in web search logs. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, pages 77–86, New York, NY, USA. ACM.

[KAN08] : Kanaan, G., Al-Shalabi, R., Ghwanmeh, S., and Bani-Ismail, B. (2008). Interactive and Automatic Query Expansion: A Comparative study with an Application on Arabic. American Journal of Applied Sciences 5, 11: 1433–1436.

[KRA04] : Kraft, R. and Zien, J. (2004). Mining anchor text for query refinement. In Proceedings of the 13th international conference on World Wide Web. ACM Press, New York, NY, USA, pp. 666–674.

[KWAN15]: Kwan, P. W., Welch, M. C., Foley, J. J., Kwan, P., Welch, M. et Foley, J. (2015). A knowledge-based decision support system for adaptive fingerprint identification that uses relevance feedback. Knowledge-Based Systems, 73(Complete):236–253.

[LATIRI12] : Latiri, C., Haddad, H. et Hamrouni, T. (2012). Towards an effective automatic query expansion process using an association rule mining approach. Journal of Intelligent Information Systems, 39(1):209–247.

[LEE08] : Lee, K. S., Croft, W. B. et Allan, J. (2008). A cluster-based resampling method for pseudorelevance feedback. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 235–242, New York, NY, USA. ACM.

[LI12]: Li, C. et Wang, J. (2012). A clustering approach to improving pseudo-relevance feedback : Improving retrieval effetiveness by removing noisy documents. 2012

Fourth International Symposium on Information Science and Engineering, 0:35–38.

[LIU04]: Liu, S., Liu, F., Yu, C. et Meng, W. (2004). An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '04, pages 266–272, New York, NY, USA. ACM.

[LV10] : Lv, Y. and Zhai, C. (2010). Positional relevance model for pseudorelevance feedback. Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '10, pp. 579-586.

[MIN10]: Min, J., Leveling, J., Zhou, D. et Jones, G. J. F. (2010). Document expansion for image retrieval. In Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO '10, pages 65–71, Paris, France, France. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

[MIT98]: Mitra, M., Singhal, A. et Buckley, C. (1998). Improving automatic query expansion. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '98, pages 206–214, New York, NY, USA. ACM.

[NAV05] : Navigli, R. And Velardi, P. (2005). Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. IEEE Trans. Patt. Anal. Mach. Intell. 27 (7):1075–1088.

[MAS13] : Massih-Reza Amini, Eric Gaussier. Recherche d'information : Applications, modèles et algorithmes. Fouille de données, décisionnel et big data. Editeur(s) : Eyrolles.

[NAV09] : Navigli, R. (2009). Word Sense Disambiguation: A Survey. ACM 0360-0300/2009/02-ART10. ACM Computing Surveys, DOI. 10.1145/1459352. 1459355. http://doi.acm.org/10.1145/ 1459352.1459355. 41(2), Article 10.

[NAWAB16]: Nawab, R., Stevenson, M. et Clough, P. (2016). An ir-based approach utilising query expansion for plagiarism detection in medline. International Journal of Computational Biology and Drug Design.

[PACKER12] : Packer, H. S., Samangooei, S., Hare, J. S., Gibbins, N. et Lewis, P. H. (2012). Event detection using twitter and structured semantic query expansion. In Proceedings of the 1st International Workshop on Multimodal Crowd Sensing, CrowdSens '12, pages 7–14, New York, NY, USA. ACM.

[ROB76] : Robertson S.E. and Sparck-Jones. J.K. (1976). Relevance weighting of search terms. Journal of the American Society for Information Science, 27(3):129- 146.

[ROB90]S.E. ROBERTSON, (1990) "ON TERM SELECTION FOR QUERY EXPANSION", Journal of Documentation, Vol. 46 Issue: 4, pp.359-364, https://doi.org/10.1108/eb026866.

[ROC71] : Rocchio, J.J. (1971). The SMART Retrieval System: Experiments in Automatic Document Processing, chapter Relevance Feedback in Information Retrieval, pp. 313–323.

[RUT03] : Ruthven, I. (2003). Re-examining the Potential Effectiveness of Interactive Query Expansion. Proceedings of ACM SIGIR'03. Toronto, Canada. ISBN:1-58113-646-3 doi.10.1145/860435.860475, pp. 213-220.

Bibliographie

[SAL97] :Salton, G. et Buckley, C. (1997). Readings in information retrieval. chapitre Improving Retrieval Performance by Relevance Feedback, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[SEG14]: Segura, A., Vidal-Castro, C., Ferreira-Satler, M., Sánchez, S. (2014). Domain Ontology-Based Query Expansion: Relationships Types-Centered Analysis Using Gene Ontology. Proceedings of the 2nd Colombian Congress on Computational Biology and Bioinformatics (CCBCOL). DOI. 10.1007/978-3-319-01568-2_27. Springer International Publishing Switzerland, pp. 183-188 .

[SON07] : Song, M., Song, I. -Y., Hu, X., and Allen, R. B. (2007). Integration of association rules and ontologies for semantic query expansion. Journal of Data and Knowledge Engineering. 63(1):63–75.

[SPA11] : Shin'ichi T., Komei S., Yuhei A., Koji Z. (2011) Spatio-Temporal Pseudo Relevance Feedback for Scientific Data Retrieval.

[THES14]: Thesprasith, O. et Jaruskulchai, C. (2014). Query expansion using medical subject headings terms in the biomedical documents. In Nguyen, N., Attachoo, B., Trawiński, B. et Somboonviwat, K., éditeurs : Intelligent Information and Database Systems, volume 8397 de Lecture Notes in Computer Science, pages 93–102. Springer International Publishing.

[UG16] : Mémoire de Master. Un système de classification de documents textuels sur le web collaboratif. Juin2016 chapitre3 page 39.

[VOO04] : Voorhees, E. (2004). Overview of the trec 2004 robust track. In Proceedings of the 13th Text REtrieval Conference (TREC-7), NIST Special Publication. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, pp. 500-261.

[VOO06] : Voorhees, E. (2006). Overview of the trec 2005 robust retrieval track. In E.M. Voorhees and L. P. Buckland, editors, The Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, MD,NIST.

[VOO93]: Voorhees, E. (1993). Using wordnet to disambiguate word senses for text retrieval. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Pittsburgh, Pennsylvania, USA, pp.171–180.

[VOO94] : Voorhees, E. (1994). Query expansion using lexical-semantic relations. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Dublin, Ireland, pp. 61–69.

[WAN09] : Wang, H., Liang, Y., Fu, L., Xue, G.-R., and Yu, Y. (2009). Efficient query expansion for advertisement search. Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press. Boston, MA, USA, pp. 51–58.

[XU96]: Xu, J. et Croft, W. B. (1996). Query expansion using local and global document analysis. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '96, pages 4–11, New York, NY, USA. ACM.

Bibliographie

[YU03]: Yu, S., Cai, D., Wen, J.-R. et Ma, W.-Y. (2003). Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In Proceedings of the 12th International Conference on World Wide Web, WWW '03, pages 11–18.

Webographie

webographie

[web1] : Disponible sur « http://ucanaccess.sourceforge.net/site.html. ». consulté le 5 Mai 2018.

[web2] : http://www.naunaute.com/liste-stop-words-francais-393. » Lionel Miraton, consulté le 21 février 2013.

[web3] : Disponible sur « https://en.wikipedia.org/wiki/Sequential_minimal_optimization. »

[web4] : Disponible sur « https://zestedesavoir.com/tutoriels/1760/un-peu-de-machine-learning avec-les-svm/. »,consulté Le 26 avril 2017.

[web5] : Disponible sur « https://www.cs.auckland.ac.nz/courses/compsci367s1c/tutorials/IntroductionToWeka.pdf. »

[web6] : Disponible sur « https://fr.wikipedia.org/wiki/Cooccurrence. ». consulté Le 19 Mai 2018

[web7] : Disponible sur « https://fr.wikipedia.org/wiki/Apprentissage_automatique. ».consulté Le 21 Mai 2018

[web8] : Disponible sur « http://fr.wikipedia.org/wiki/Java_(langage). » consulté le 21 septembre2017.

[web9] : Disponible sur « https://fr.wikipedia.org/wiki/NetBeans. consulté en octobre 2016. »

[web10] : Disponible sur « http://www.plantie.fr/EMA/projetdatamining/fouille2010-1/docs/weka1.pdf. »