Table des matières

Liste des figures

Liste des Tableaux

Introduction générale

Introduction générale

Face à l'évolution rapide des moyens d'information et de communication, une masse importante d'information est produite tous les jours à travers des milliers de livres et d'articles de journaux. Cette forte augmentation quantitative d'information à gérer et à consulter ne peut plus se contenter des méthodes et des techniques classiques de stockage et de consultation utilisées jusqu'ici.

Dans ce contexte, les Systèmes de Recherche d'Information (SRI) sont développes en vue d'automatiser la gestion de ces informations et restituer a un utilisateur l'information qu'il recherche, le plus facilement et le plus rapidement possible.
A cet effet, ils sont dotés de fonctionnalités de stockage et d'organisation des informations, ainsi que des fonctionnalités de recherche et de restitution des documents susceptibles de répondre à une demande donnée.

De ce fait, des approches de recherche d'information, regroupant entre autres des techniques d'indexation ainsi que des mécanismes d'appariement et de reformulation ont été développées afin de mieux répondre aux besoins de l'utilisateur.

Les premières approches de recherche d'information, qualifiées de classiques, se basent sur une recherche par mots clés, les documents sont représentés comme des sacs de mots souvent pondérés, et la pertinence d'un document vis-à-vis d'une requête est souvent estimée en s'appuyant sur les fréquences d'apparition des mots de la requête dans ces mêmes documents.

Les SRI sont alors confrontés à un nouveau défi du à ces limites.
Ce qui a poussé des chercheurs à marquer un arrêt pour explorer d'autres terrains, notamment celui de la linguistique et de l'intelligence artificielle, pour proposer des améliorations à la solution, qui est généralement préconisée aujourd'hui.

Introduction générale

Les moteurs de recherche permettent de retrouver des ressources associées généralement à des mots clés résultant d'une phase d'annotation. L'attribution de mots clés à un document est une tâche difficile, voire impossible à effectuer manuellement, d'une part à cause du temps et de l'effort que cela représente, et d'autre part à cause du fait que le résultat d'une annotation peut varier d'un expert a un autre.

L'utilisation d'annotation est motivée par plusieurs raisons. On peut citer par exemple la difficulté d'évaluer les performances d'un SRI sans un corpus annoté permettant ainsi de calculer différents paramètres comme le rappel et la précision.

Problématique

La recherche d'information est un processus qui se base essentiellement sur la requête exprimé par l'utilisateur pour répondre à ses besoins. En effet, le résultat d'une recherche ne peut-être pertinent si la requête ne d écrit pas explicitement et clairement les besoins de l'utilisateur ainsi que la bonne représentation des informations apparus dans la base de collection des documents(Corpus). Cela, issu au mal compréhension du domaine recherché ou à une limitation des connaissances d'utilisateur.

Les travaux décrits dans ce mémoire s'intéressent à l'annotation d'un corpus pour l'évaluation des systèmes de recherche d'informations.

Objectifs

Dans le cadre de ce mémoire, on se propose d'apporter une solution aux problèmes d'évaluation des SRI en proposant un outil pour l'annotation des corpus. Notre travail se décompose en plusieurs palais :

Un SRI doit être bien évalué, sinon il peut demeurer pratiquement inexploitable et impossible à utiliser. L'annotation des documents est une solution à ce problème. Notre contribution consiste à développer un outil pour l'annotation des documents.

Organisation du mémoire

Ce mémoire est composé de trois chapitres :

- Le premier chapitre présente les notions de base et les principaux concepts utilisés dans le domaine de la recherche d'information. Il présente l'architecture générale d'un système de recherche d'information (SRI) telle qu'elle est admise actuellement , il comporte aussi les différentes mesures d'évaluation de pertinence.

- Le deuxième chapitre dresse un état d'analyse et Indexation des documents et des requêtes, et les Approches possibles pour réaliser un système de RI
  L'objectif de l'analyse et de l'indexation est trouver des concepts les plus importants dans les documents et créer une représentation interne en utilisant ces concepts. Il se termine par l'évaluation d'un SRI et Comment évaluer Précision-Rappel ? dans le domaine de la recherche d'information.

- Le troisième chapitre est le cœur de notre mémoire. Il concerne l'application du concept d'annotation d'un corpus. Il décrit notre modélisation et implémentation de notre outil pour l'annotation des corpus.

Dans la conclusion, nous présenterons les principaux résultats obtenus dans ce mémoire ainsi que quelques perspectives.

Chapitre I :

Recherche d'information

- concepts de base -

1. Introduction

Le monde assiste depuis ces dernières décennies, a une production massive d'informations dans tous les domaines d'intérêt. De multiples directions de recherche ont tenté de mettre en œuvre des processus automatiques d'accès à l'information. L'objectif est d'exploiter au mieux les bases volumineuses de ces informations.

Un Système de Recherche d'Information (SRI), nécessite la combinaison de modèles et algorithmes. Ces derniers permettent la représentation, le stockage, la recherche et la visualisation des informations. L'objectif principal de ce système est de mettre en œuvre un processus de comparaison entre besoin utilisateur et documents d'une collection dans le but de retrouver ceux qui sont pertinents. L'élaboration d'un mécanisme de recherche d'information pose alors des problèmes lies tant à la représentation qu'a la localisation de l'information pertinente. En effet, la recherche d'information induit un processus d'inférence véhicule par l'objet de la requête, en se basant sur une description structurelle des unîtes d'information.

Tout au long de ce chapitre nous allons passer en revue les concepts, les approches utilisés dans le domaine des SRI, notre intérêt se porte ainsi sur les principes de la recherche d'information.; nous présentons d'une part les mesures utilisées pour comparer les performances des SRI et d'autre part les collections de tests largement utilisées dans le domaine de la recherche d'information.

2. Généralités sur les Systèmes de Recherche d'Information (SRI)

ü 2.1 Définition

La recherche d'information [Frakes 1992] [Grossman et al 1998] [Salton 1970] [Salton 1989] [Yates et al 1999] est l'ensemble des techniques permettant de gérer des textes. Gérer des textes ou des documents implique stocker, rechercher et explorer des documents pertinents.

Un système de recherche d'information intègre un ensemble de techniques et de processus permettant de sélectionner dans une collection de documents ceux qui sont susceptibles de répondre au besoin d'un utilisateur. Ces processus permettent :

- ü la représentation des informations et des besoins,
- ü l'interrogation, la recherche et la sélection des informations pertinentes répondant aux besoins d'un utilisateur.

La problématique majeure émanant de tout système de recherche d'information est de retrouver les quelques dizaines ou milliers de documents pertinents parmi des millions de documents. Cet écart de cardinalité rend cette tache encore plus difficile.

ü 2.2 Concepts clés de la recherche d'information

Un système de recherche d'information, intègre un ensemble de modèles pour la représentation des unites d'informations (documents et requêtes). Il intègre également un mécanisme de recherche/sélection. Ce dernier permet de sélectionner l'information pertinente en réponse aux besoins exprimés par l'utilisateur à l'aide d'une requête.

Il peut être représente par le processus en U de recherche d'information [Hlaoua, 07].

La figure 1.1 illustre l'architecture générale d'un système de recherche d'information.

Plusieurs éléments clés y sont distingués :

- la collection de documents,
- les documents,
- le besoin en information (requête),
- la représentation des documents et des requêtes (indexation ou analyse),
- l'appariement requête-document,
- la reformulation automatique de requêtes

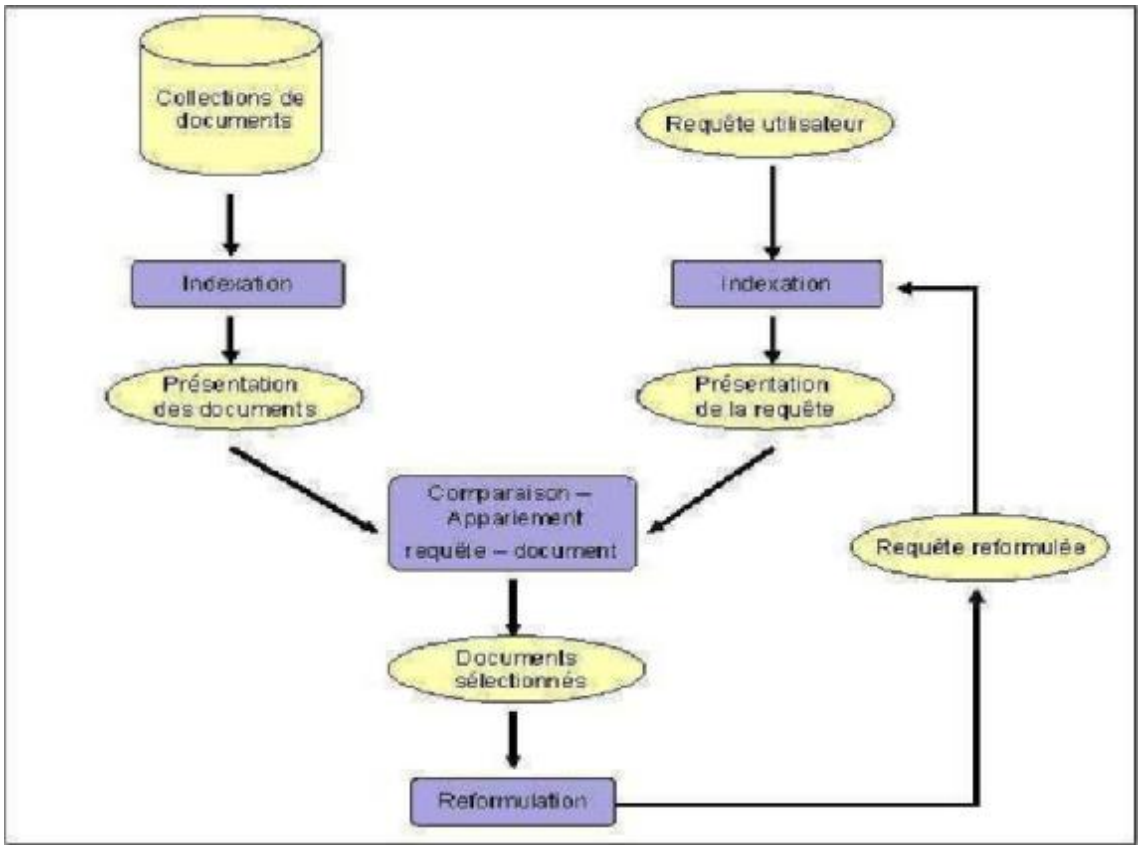


Figure I.1 : Le Processus en U de la Recherche d'Information [Hlaoua, 07]

Ces traitements sont de deux catégories : la représentation (indexation) et la recherche. Dans la section qui suit, nous allons définir ces éléments séparément.

§ 2.2.1 La collection de documents

La collection de documents constitue l'ensemble des informations exploitables et accessibles par l'utilisateur. Elle est constituée d'un ensemble de documents. Dans le cas général et pour des raisons d'optimalité, la collection constitue des représentations très simplifiées mais suffisantes de ces documents. Ces représentations sont étudiées de telle sorte que la gestion (ajout, suppression d'un document) et l'interrogation (recherche) de la collection se font dans les meilleures conditions de cout.

§ 2.2.2 Le document

Le document constitue l'information élémentaire d'une collection documentaire. L'information élémentaire, appelée aussi granule de document, peut représenter tout ou une partie d'un document. Dans la suite de ce mémoire, nous utiliserons indifféremment les termes document ou information pour désigner un granule documentaire.

§ 2.2.3 Le besoin en information

Un besoin en information d'un utilisateur est exprimé par une requête. La littérature propose divers types de langages d'interrogation pour formuler cette requête. Nous citons les plus répandus :

– interrogation en langage booléen,

– interrogation en langage naturel ou quasi naturel,

– interrogation en langage graphique.

Détaillons à présent ces différents langages.

1. ***Interrogation en langage booléen :***

   L'utilisateur exprime sa requête sous forme d'un ensemble de termes relies entre eux par des opérateurs booléens (ET, OU, NON). Beaucoup de moteurs de recherche, se basent sur ce mode d'interrogation, citons les plus connus : Altavista, Google, etc.

2. ***Interrogation en langage naturel ou quasi naturel :***

   L'utilisateur exprime sa requête en langage libre (langage naturel) sous forme de mots clés. Le système se charge de traduire (analyser) ces mots clés en une requête de langage de base de données ou une autre forme interne utilisable par le système.

Les systèmes SMART [Salton 1989], SPIRIT [Fluhr et al 1985], OKAPI
[Robertson et al 1976] et MercureO2 [Boughanem 1992] sont interrogeables en langage naturel.

3. ***Interrogation en langage graphique :***

   Une interface d'aide à la formulation de la requête est proposée à l'utilisateur. En effet, une vue d'ensemble de la base d'information et en particulier une vue de termes représentant le contenu sémantique des documents, est donnée a l'utilisateur pour l'assister à formuler sa requête.

   Dans PROTEUS [Signore et al 1992], l'interface d'aide à la formulation de requête propose un gestionnaire de thesaurus. Ce dernier est représenté par un graphe, les nœuds étant les termes du thesaurus et les liens étant les relations sémantiques entre ces termes. L'utilisateur peut identifier le type de relation qu'il souhaite utiliser et sélectionner un terme

   Le projet NEURODOC [Lelu et al 1992] est plus adapte à l'utilisation d'un thesaurus volumineux. NEURODOC offre à l'utilisateur un tableau de bord ou

chaque nœud possède un nom et résume le sous-ensemble de mots et de documents fortement lies.

§ 2.2.4 La représentation des documents et des requêtes (indexation ou analyse) :

La représentation des documents et des requêtes est supportée par un ensemble de règles et notations permettant la traduction d'une requête ou d'un document d'une description brute vers une description structurée.

Ce processus de conversion est appelé ***Indexation***

L'indexation est une opération permettant d'extraire d'un document ou d'une requête une représentation paramétrée qui couvre au mieux son contenu sémantique.

Le résultat de l'indexation constitue le descripteur du document ou de requête.

Ce dernier est souvent une liste de termes ou groupe de termes significatifs pour l'unité textuelle correspondante, généralement assortis de poids représentant leur degré de représentativité du contenu sémantique de l'unité qu'ils décrivent.

Les descripteurs des documents (mots, groupe de mots) forment le langage d'indexation. L'indexation, est une étape primordiale dans la recherche d'information. De sa qualité dépend en partie la qualité des réponses du système.

– Indexation manuelle :

Dans le cas de l'indexation manuelle, chaque document est analyse par un spécialiste du domaine ou par un expert documentaliste.

En fonction de ses connaissances, cet expert détermine, les mots clés qui lui semblent les plus significatifs pour représenter le document. L'indexation humaine est une activité fondée sur le jugement d'un être humain. Elle se caractérise par sa profondeur, sa cohérence (ce qui est fondamental pour la cohérence du fond et des fichiers) et sa qualité (exhaustivité - spécificité). Elle est cependant trop dépendante de l'état des connaissances des indexeurs. Cela induit à la subjectivité de ses résultats. Elle nécessite la lecture de l'intégralité des documents. Son application est de ce fait inadaptée à des collections de taille importante. Les collections TREC1 constituent un exemple significatif. Elles contiennent des millions de documents extraits d'Internet (le web).

L'indexation automatique permet de pallier à ce problème.

– Indexation automatique :

L'indexation automatique reconnait des chaines de caractères constitutives de mots non vides. Elle détecte automatiquement les termes les plus représentatifs du contenu du document. Ce type d'indexation est actuellement la méthode la plus répandue.

Elle comprend deux étapes fondamentales : l'identification des termes d'indexation et l'évaluation de leurs poids.

L'identification des termes d'indexation consiste à analyser le texte du document mot à mot. Son objectif est d'en extraire les mots vides qui ne jouent qu'un rôle syntaxique. Ces mots sont identifiés puis élimines grâce à un anti dictionnaire (stoplist en Anglais). Les mots apparaissant trop souvent n'ont aucun intérêt. Ils sont également élimines. Seuls les mots significatifs représentant les concepts du document sont retenus.

Afin d'augmenter la qualité de la recherche, la pondération des termes extraits est primordiale. Pour mettre en évidence les diverses contributions d'un terme dans la représentation d'un document un poids lui est attribué.

§ 2.2.5 L'appariement requête-document

Le processus d'appariement requête-document est le noyau d'un système de recherche d'information. Il permet d'associer à chaque document une valeur de pertinence vis à vis d'une requête. Les documents ayant une pertinence positive sont sélectionnés.

La mesure de pertinence est calculée à partir d'une fonction de similarité, notée *RSV* (*Q,d*)(*RetrievalSattusV alue*), *Q* étant une requête et *d* un document.

Elle tient compte des poids des termes déterminés en fonction d'analyses statiques et probabilistes. Notons que ce processus est étroitement lié aux représentations des documents et des requêtes.

– la pertinence utilisateur :

Le document est juge pertinent par l'utilisateur en fonction de son besoin en information,

– la pertinence système :

Le document est juge pertinent par le SRI pour une requête sur la base de la fonction de pertinence.

ü 2.3 Reformulation automatique de requêtes

La reformulation automatique de requêtes permet de générer une requête plus adéquate à la recherche d'information dans l'environnement du SRI, que celle initialement formulée par l'utilisateur. Son principe est de modifier la requête de l'utilisateur par ajout de termes significatifs et/ou par ré-estimation de leur poids.

3. Conclusion

Dans ce première chapitre nous avons présenté les principales notions et concepts de base de la recherche d'information **.**

Dans le chapitre qui suit, nous présentons les Approches possibles pour réaliser un système de SRI, et d'Analyse et Indexation des documents et des requêtes ainsi évaluation des performances des systèmes de recherche d'information.

Chapitre II:
Analyse et Indexation des documents

1. Introduction :

Objectif

L'objectif de l'analyse et de l'indexation est :

- **trouver des** concepts les plus importants dans les documents,
- créer une représentation interne en utilisant ces concepts.

Pour trouvé des concepts il est nécessaire de procéder une analyse sémantique pour déterminer ce qui est un concept dans un texte. Cette analyse n'est pas disponible pour la RI. Les techniques existantes sont souvent restreintes à un domaine très spécialisé, et l'analyse est très complexe. *Ainsi, en pratique,* On cherche plutôt des *représentants* des concepts. Ces représentants peuvent être de formes différentes:

- des mots simples,
- des termes (éventuellement composés),
- ou des doublets de mots (groupes de deux mots).

La question qui se pose est quels représentants choisir? La réponse est que cela dépend de deux critères essentiellement: la facilité de traitement et la précision de représentation de sens.

L'idée d'utiliser des mots comme des représentants de concepts est assez naturelle.

Les mots sont des unités linguistiques qui sont les plus faciles à reconnaître et assez porteuses de sens. Ce sont ces unités qu'on utilise le plus souvent dans les systèmes actuels. Les mots ne donnent pas une description toujours très précise.

Exemple,

Le concept de "recherche d'information", est représenté par : Les mots "recherche" et "information", (Perd beaucoup de sens, car les mots "recherche" et "information" sont très courants en français, et ils ont des sens très imprécis.)

Une solution :

Proposer des approches visant à regrouper des mots pour former des termes composés.

(Ces approches utilisent soit une analyse syntaxique et/ou statistique, soit un dictionnaire de termes composés.) . Les représentants sont aussi appelés des **index**,

2. Approches possibles :

Approches possibles pour réaliser un système de RI.

2.1 Une première approche très naïve consiste à considérer une requête comme une chaîne de caractères, et un document pertinent comme celui qui contient cette chaîne de caractères.

À partir de cette vision simpliste, on peut imaginer l'approche qui consiste à balayer les documents séquentiellement, en les comparants avec la chaîne de caractères qui est la requête. Si on trouve la même chaîne de caractère dans un document, alors il est sélectionné comme réponse.

- Cette approche est très simple à réaliser.
- Elle a plusieurs lacunes

- **Vitesse:** L'opération de recherche est **très lente**. Il n'est donc envisageable d'utiliser cette approche que sur des collections très petites jusqu'à **quelques centaines de documents**.

- **Pouvoir d'expression d'une requête:** Une requête étant une simple chaîne de caractères, il est difficile d'exprimer des besoins complexes comme "Trouver des documents concernant la base de données et l'intelligence artificielle utilisées dans l'industrie".

La plupart de systèmes existants utilisent une approche différente basée sur une ***Indexation***.

2.2 . L'approche basée sur une indexation

Dans cette approche, on effectue certains prétraitements sur les documents et les requêtes, ce qu'on appelle l'indexation.

La structure d'index est de la forme suivante (la structure de fichier inversé) :

Mot → {…, Doc, …}

C'est-à-dire, chaque mot est mis en correspondance avec les documents qui le contiennent.

Une requête peut être maintenant une expression plus complexe, incluant des opérateurs logiques (ET,OU, …) ou d'autres types d'opérateurs.

L'évaluation est compositionnelle, c'est-à-dire, on commence par évaluer les éléments de base (par exemple, des mots) dans la requête, obtenant ainsi des listes de documents;

Ensuite, on combine ces listes selon l'opérateur qui relie ces éléments pour obtenir finalement une seule liste de documents.

Cette approche a les avantages suivants:

- Elle est **plus rapide**. En effet, on n'a plus besoin du parcours séquentiel. Avec la structure d'index, on peut directement savoir quels documents contiennent tel ou tel mot.
- L'expression des requêtes peut être très complexe, exprimant des besoins d'information complexes.

Le prix à payer pour ces avantages est le besoin de l'espace de stockage supplémentaire pour la structure d'index. En général, cet espace correspond de 40% à
200% de la taille de collection de documents, selon la complexité de l'indexation.

Mais ce besoin d'espace pose de moins en moins de problème maintenant.

En utilisant cette deuxième approche, on peut voir les opérations et l'environnement de la RI comme suit:

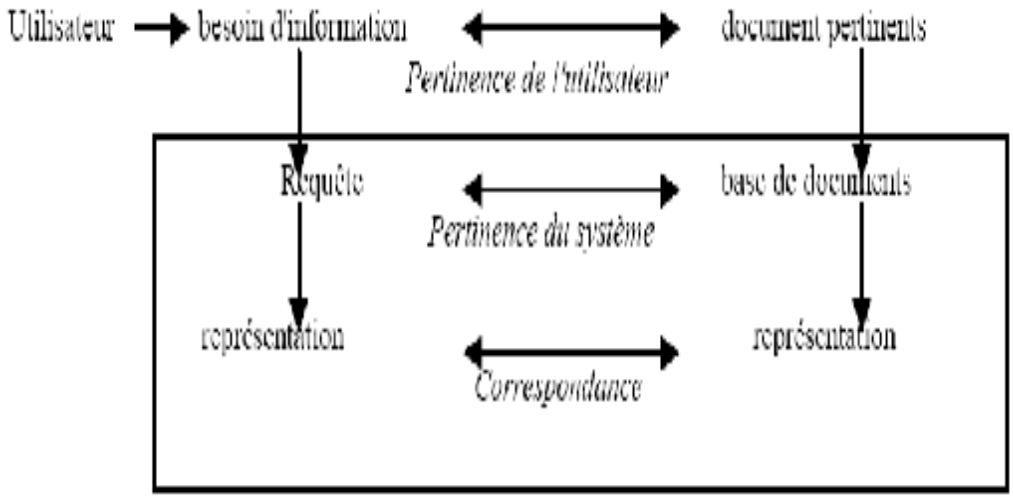


Figure II.2 : opérations et l'environnement de la RI

2.3 Approche basée sur la fréquence d'occurrences

2.3.1. Approche I : Consiste à choisir les mots représentants selon leur fréquence d'occurrence. La façon la plus simple consiste à définir un seuil sur la fréquence: si la fréquence d'occurrence d'un mot dépasse ce seuil, alors il est considéré important pour le document.

Remarque

Les mots les plus fréquents sont des mots fonctionnels (ou mots outils, mots vides). En français, les mots "de", "un", "les", etc. sont les plus fréquents. En anglais, ce sont "of", "the", etc. En arabe, ce sont ' من ' , ' حتى ', ' كلما ', etc.

La loi de Zipf:

Rang	Mot	Fréquence	Rang * Fréquence
1	The	69 971	69 971
2	Of	36 411	72 822
3	And	28 852	86 556
4	To	26 149	104 596
5	A	23 237	116 185
6	In	21 341	128 046
7	That	10 595	76 165

Tableaux II.1 : représente La loi de Zipf

Si on classe les mots dans l'ordre décroissant de leur fréquence, et on leur donne un numéro de rang (1, 2, …),

Alors : Rang * fréquence ≅ constante.

Selon cette loi, la distribution de mots suit la courbe suivante:

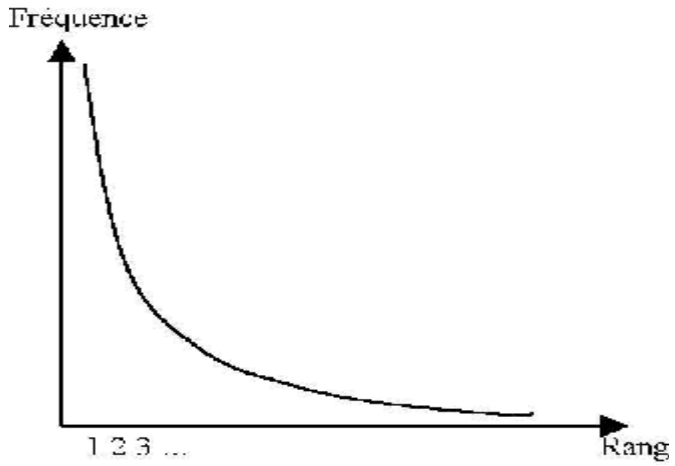


Figure II- 3 : courbe de distribution de mots Selon La loi de Zipf

Il devient évident qu'on ne peut pas garder tous les mots les plus fréquents comme des index.
Donc, on peut définir un autre seuil maximal:
Si la fréquence d'un mot dépasse ce seuil, alors il n'est pas considéré comme index.
L'utilisation de ces deux seuils correspond à ce qu'on croit sur ***l'informativité*** de mot.

2.3.2. Définition *l'informativité*

L'informativité mesure la quantité de sens qu'un mot porte.

Cette notion :

- n'est pas définie très précisément dans la RI,
- elle est utilisée seulement de façon intuitive.

La correspondance entre ***l'informativité*** et la ***fréquence*** est illustrée dans la figure suivante:

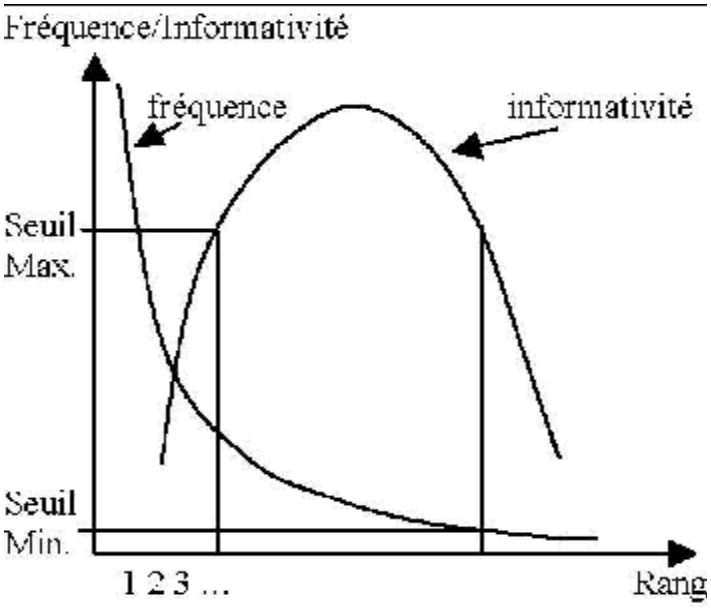


Figure II. 4 : La correspondance entre ***l'informativité*** et la ***fréquence***

En choisissant les mots qui ont des fréquences entre les deux seuils, on espère obtenir les mots dont l'informativité est la plus élevée.

2.4 Approche basée sur la valeur de discrimination

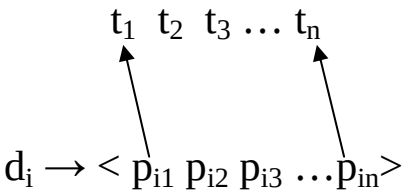
Par ***"discrimination",*** on réfère au fait qu'un terme distingue bien un document des autres documents.

- un terme qui a une valeur de discrimination élevée doit apparaître seulement pour un petit nombre de documents.
- Un terme qui apparaît dans tous les documents n'est pas discriminant.

L'idée est de garder seulement les termes discriminants, et éliminer ceux qui ne le sont pas.

Le calcul de la valeur de discrimination a été développé dans le modèle vectoriel.

Dans le modèle vectoriel, chaque document est représenté par un vecteur de poids comme suit:



Où

P_{ij} : est le ***poids*** du terme t_j dans le document d_i .

Le calcul de la valeur de discrimination d'un terme se fait comme suit:

L'idée est que, si on uniformisant le poids d'un terme dans tous les documents, on obtient une grande amélioration dans l'uniformité du corpus, ce terme était donc très différent (non uniformément distribué) dans différents documents. Il a donc une grande valeur de discrimination. En revanche, si en uniformisant le poids du terme, on n'obtient pas beaucoup d'amélioration sur l'uniformité, ce terme était donc déjà distribué de façon uniforme, donc peu discriminant.

1) On calcule d'abord le vecteur centroïde (ou le vecteur moyen) du corpus comme suit:

Pour chaque terme, son poids dans le vecteur centroïde ***V*** est le poids moyen de ses poids dans les documents. C'est-à-dire:

$$\mathbf{p_j = \Sigma_i\, p_{ij} / N}$$

Où **N** est le nombre de documents dans le corpus.

2) On calcule l'uniformité du corpus comme la similarité moyenne des documents avec le centroïde:

$$\mathbf{U_1 = C * \Sigma j\ Sim\ (di, V)}$$

Où **C** est une constante de normalisation (par exemple **1/N**),

Et **Sim (di, V)** est la similarité entre le document **di** et le vecteur centroïde **V**.

Sim doit être une formule normalisée qui donne une valeur dans **[0,1]**

3) On uniformise le poids du terme en question à **0**, et on répète les deux étapes ci-dessus pour obtenir une nouvelle valeur d'uniformité **U2.**

4) La valeur de discrimination du terme est:

$$\mathbf{V = U2 - U1.}$$

Dans ce calcul de la discrimination, on ne se préoccupe pas beaucoup de la fréquence d'un terme dans un document particulier, mais beaucoup plus à sa distribution dans le corpus.

En utilisant la valeur de discrimination, on peut éliminer les mots fonctionnels comme "de", "à", etc. qui apparaissent dans tous les documents en français.

*2.5 Approche basée sur tf*idf*

Le nom ***tf * idf*** est très connu dans le milieu de la RI.

Cela désigne un ensemble de schémas de pondération (et de sélection) de termes.

- ***tf :*** " ***t***erm ***f***requency"
- ***idf :*** "***i***nverted ***d***ocument ***f***requency".

Par ***tf***, on désigne une mesure qui a rapport à l'importance d'un terme pour un document.

En général, cette valeur est déterminée par la fréquence du terme dans le document.

Par ***idf***, on mesure si le terme est discriminant (ou non-uniformément distribué).

On donne quelques formules de ***tf*** et d'***idf*** souvent utilisées.

1. ***tf*** = fréquence d'occurrence du terme dans un document ***f(t,d);***

   tf = f(t,d) / Max[f(t,d)] où Max[f(t,d)] : fréquence maximale des termes dans d;

   tf = log(f(t,d))

   tf = log(f(t,d) + 1)

2. ***idf*** = log(N/n) où N est le nombre de documents dans le corpus, et n ceux qui contient le terme
3. Finalement, on peut aussi imposer certaine normalisation sur les valeurs calculées.

Une formule de ***tf*idf*** est donc la multiplication d'une tf par une idf.

Exemple:

tf*idf = [f(t,d) / Max[f(t,d)]] * log(N/n)

Une formule ***tf*idf*** combine les deux critères qu'on a vus:

1. **l'importance du terme** pour un document (par tf),
2. **Le pouvoir de discrimination** de ce terme (par idf).

Ainsi, un terme qui a une valeur de ***tf*idf*** élevée doit être à la fois important dans ce document, et aussi il doit apparaître peu dans les autres documents.

C'est le cas où un terme correspond à une caractéristique importante et unique d'un document.

Avec une telle formule, on peut donc choisir à garder seulement les termes dont la valeur de tf*idf **dépasse certain seuil.**

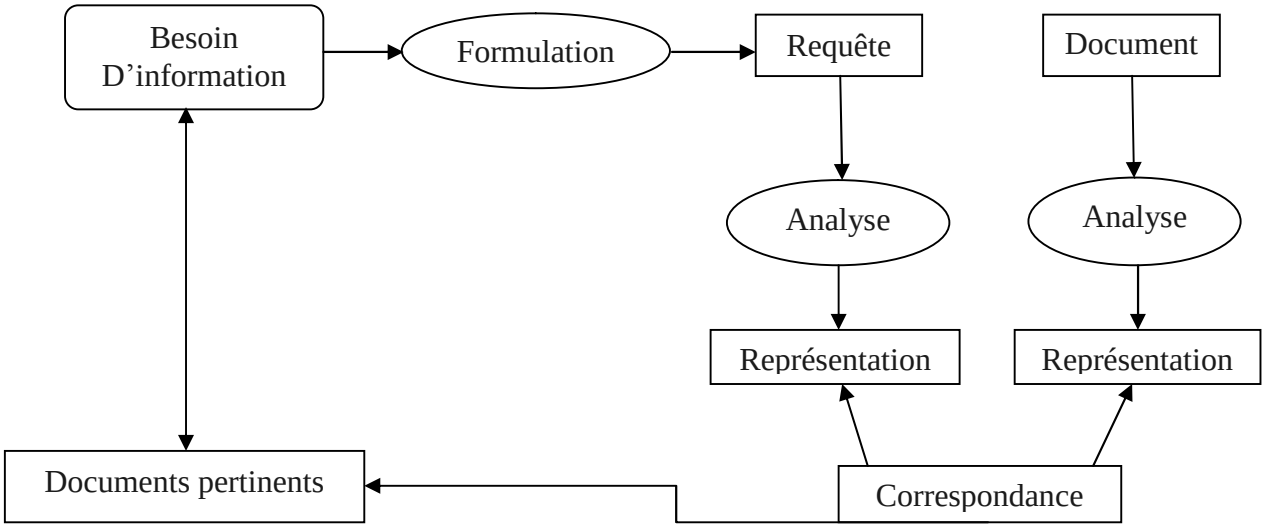


Figure II.5 : Approche classique de la RI

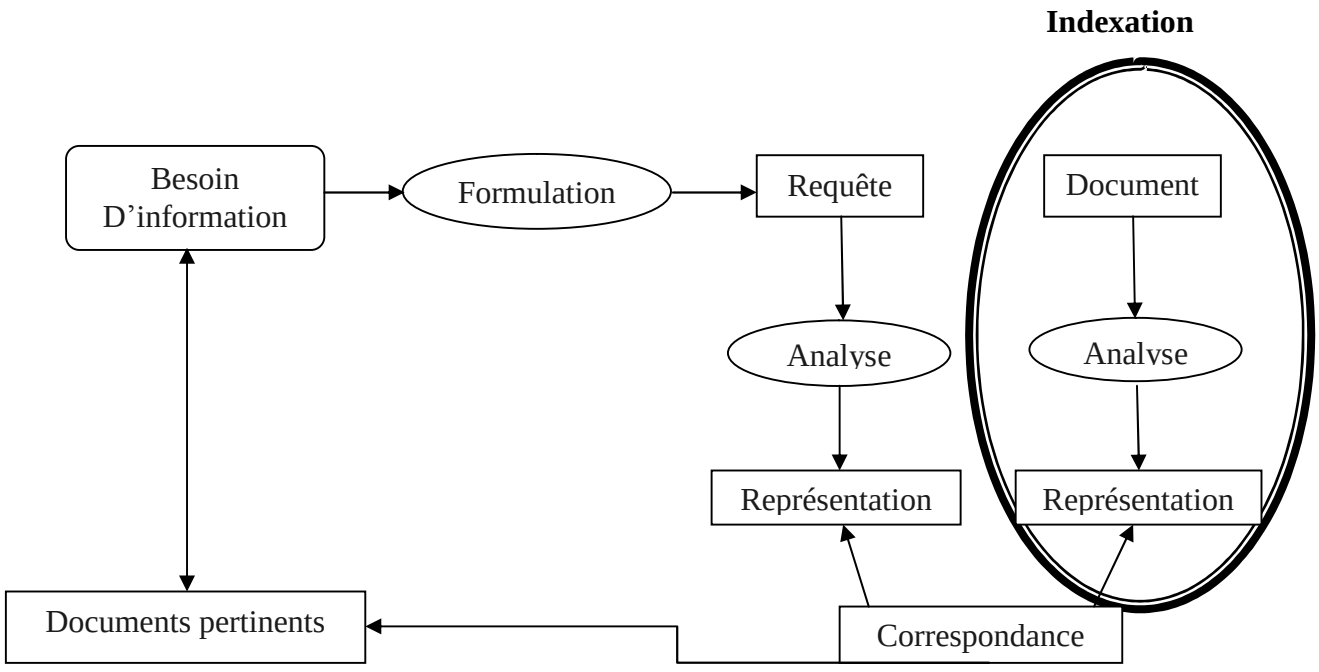


Figure II. 6 : Approche classique de la RI (Indexation)

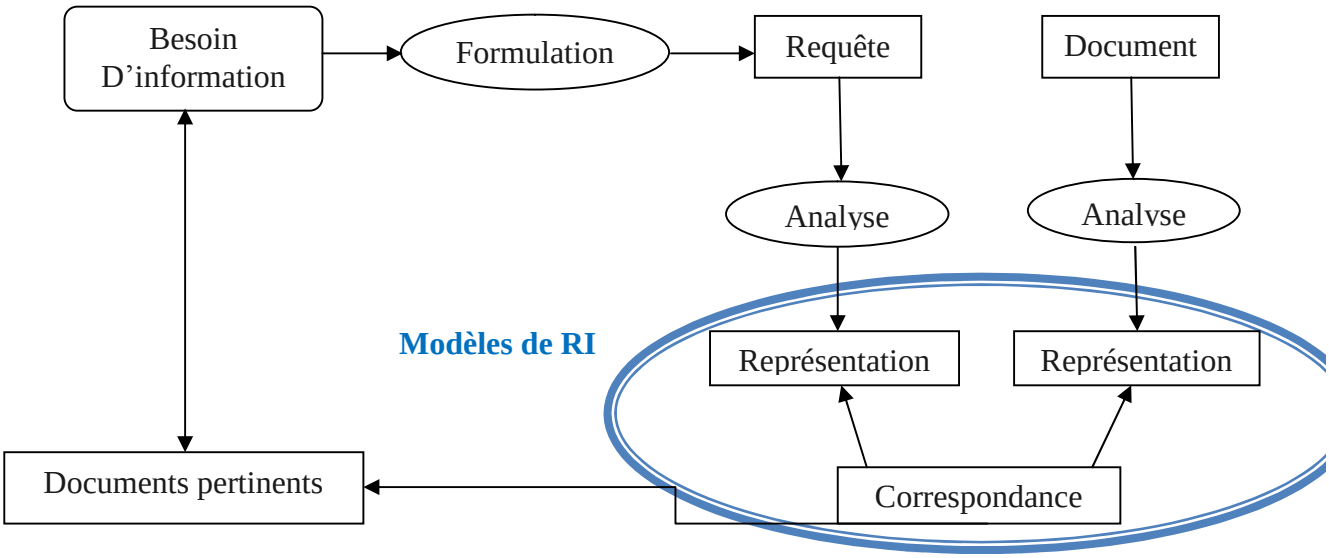


Figure II.7 : Approche classique de la RI (Modèles de RI)

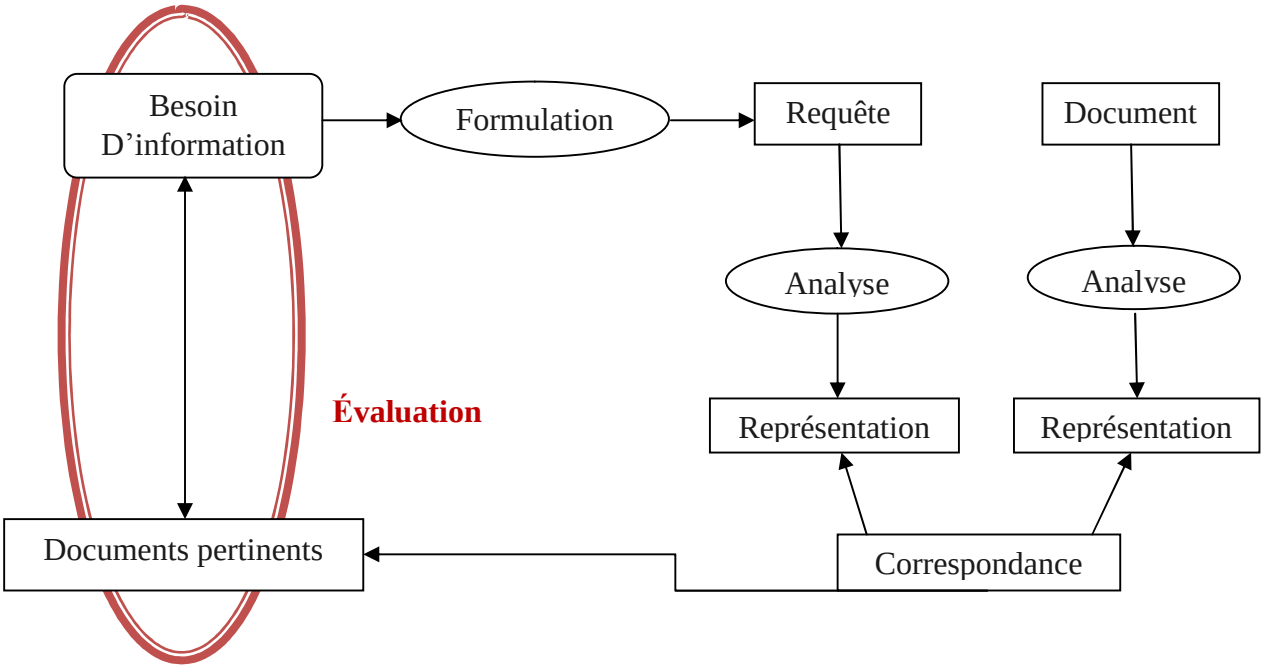


Figure II.8 : Approche classique de la RI (Évaluation)

3. La pondération de termes

La pondération d'un terme peut être de diverses natures.

- **Ø Elle peut être:**
- simplement la fréquence d'occurrence,
- ou bien une mesure dérivant de cette fréquence (par exemple, normalisée).
- Elle peut être également une formule de **tf*idf**.

Des comparaisons ont montré qu'en utilisant seulement la fréquence d'occurrence ne donne pas une performance satisfaisantes (même si on élimine les mots fonctionnels d'une certaine façon).

En général, les formules de **tf*idf** donnent de meilleures performances.

4. Le résultat de l'indexation

Durant l'indexation, on doit :

- transformer les mots (lemmatisation),
- sélectionner un ensemble d'index
- et les mesurer.

Le résultat d'une indexation est donc un ensemble de ***terme*** qui peut être :

- soit un ***mot***,
- soit une ***racine de mot,***
- soit un ***terme composé*** si on possède un mécanisme pour reconnaître des termes composés.

$$\mathbf{d \rightarrow \{ \ldots (ti, pi), \ldots \}}$$

où ***ti*** est un terme, et ***pi*** est son poids.

Cet ensemble de termes pondérés sera utilisé pour constituer une représentation du contenu du document.

5. Evaluation d'un système

La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir.

Plus les réponses du système correspond à celles que l'utilisateur espère, mieux est le système.

5.1. *Corpus de test (références)*

Pour arriver à une telle évaluation, on doit connaître d'abord les réponses idéales de l'utilisateur.

Ainsi, l'évaluation d'un système s'est faite souvent avec certains corpus de test.

Dans un corpus de test, il y a:

- un ensemble de documents;
- un ensemble de requêtes;
- la liste de documents pertinents pour chaque requête.

NB

Pour qu'un corpus de test soit significatif, il faut qu'il possède un nombre de documents assez élevé.

Les premiers corpus de test développés dans les années 1970 renferment quelques milliers de documents.

Les corpus de test plus récents (par exemple, ceux de TREC) contiennent en général plus 100 000 documents (considérés maintenant comme un corpus de taille moyenne), voir des millions de documents (corpus de grande taille).

L'évaluation d'un système ne doit pas se reposer seulement sur une requête.

Pour avoir une évaluation assez objective, un ensemble de quelques dizaines de requêtes, traitant des sujets variés, **est nécessaire**.

L'évaluation du système doit tenir compte des réponses du système pour toutes ces requêtes.

Finalement pour établir les listes de documents pour toutes les requêtes, les utilisateurs (ou des testeurs simulant des utilisateurs) doivent examiner chaque document de la base de document, et juger s'il est **pertinent**.

Après cet exercice, on connaît exactement quels **documents** sont **pertinents** pour chaque **requête**.

Pour la construction d'un corpus de test, les jugements de pertinence constituent la tâche la plus difficile.

5.2. Précision et rappel

La comparaison des réponses d'un système pour une requête avec les réponses idéales nous permet d'évaluer les deux métriques suivantes:

5.2.1. **Taux de précision** :

La précision mesure la capacité du système de rejeter tous les documents non pertinents à une requête. Il est donne par le rapport entre l'ensemble des documents sélectionnes pertinents et l'ensemble des documents sélectionnes.

La précision mesure la proportion de documents pertinents retrouvés parmi tous les documents retrouvés par le système.

$$\text{Précision} = \frac{\#\text{documents pertinents retrouvés}}{\#\text{documents retrouvés}}$$

5.2.2. **Bruit** :

B = Nombre de documents non pertinents extraits

Nombre de documents extraits

Bruit = 1− Précision

5.2.3. **Taux de rappel** :

Le rappel mesure la capacité du système à retrouver tous les documents pertinents répondants à une requête. Il est donne par le rapport entre les documents retrouves pertinents et l'ensemble des documents pertinents de la base.

$$\text{Rappel} = \frac{\#\text{documents pertinents retrouvés}}{\#\text{documents pertinents dans la base}}$$

5.2.4. **Silence** :

S = Nombre de documents pertinents non extraits

Nombre de documents pertinents

Silence = 1−Rappel

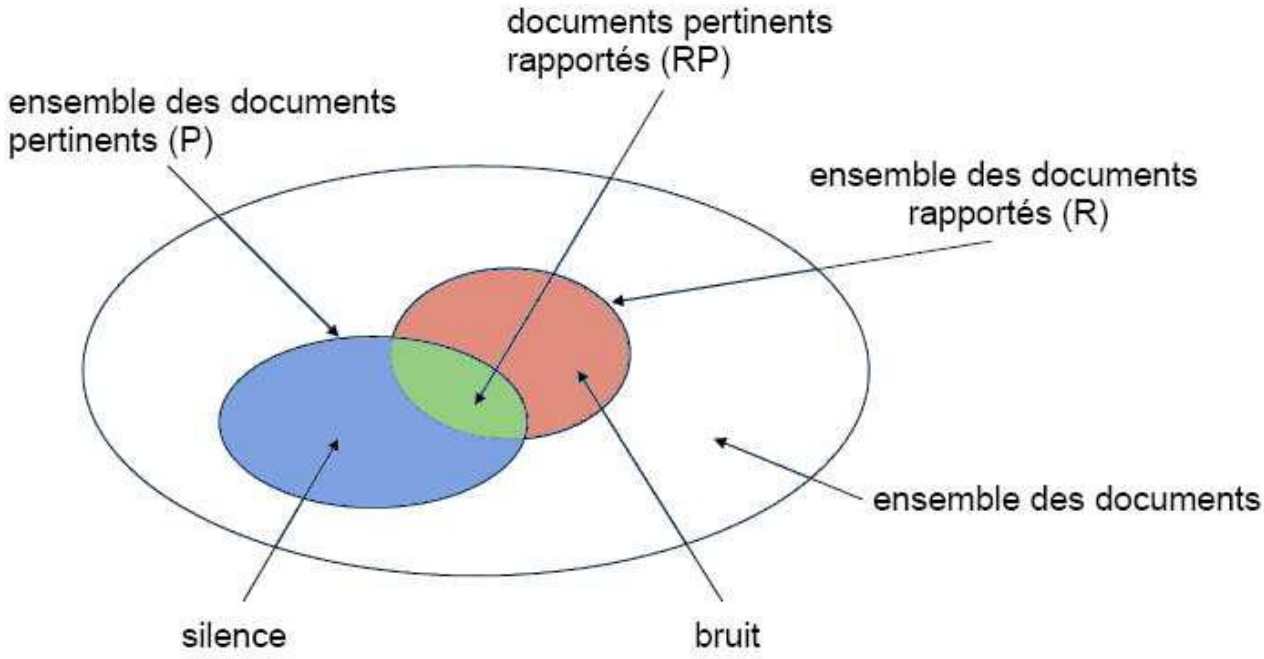


Fig. II.9 – Exemple de rappel et de précision pour une requête

La figure 9 illustre la précision et le rappel d'une requête d'une façon générale.
Toutefois, seule une partie des documents restituée par le système est examinée par l'utilisateur. Dans ce cas, la paire des mesures (taux de rappel, taux de précision) est calculée a chaque point de rappel (document pertinent restitue). Il s'agit de considérer la liste ordonnée des documents évalues, de calculer pour chaque document sélectionne la précision et le rappel, puis exprimer en fonction des valeurs trouvées la précision en fonction du rappel.
Avec ces valeurs, on trace une courbe représentant la précision en fonction du rappel.
Idéalement, on voudrait qu'un système donne de bons taux de précision et de rappel en même temps.
Un système qui aurait 100% pour la précision et pour le rappel signifie qu'il trouve tous les documents pertinents, et rien que les documents pertinents.
Cela veut dire que les réponses du système à chaque requête sont constituées de tous et seulement les documents idéaux que l'utilisateur a identifiés.
En pratique, cette situation n'arrive pas.
Plus souvent, on peut obtenir un taux de précision et de rappel aux alentours de 30%.

Les deux métriques ne sont pas indépendantes. Il y a une forte relation entre elles: quand l'une augmente, l'autre diminue.

Il ne signifie rien de parler de la qualité d'un système en utilisant seulement une des métriques.

En effet, il est facile d'avoir 100% de rappel: il suffirait de donner toute la base comme la réponse à chaque requête. Cependant, la précision dans ce cas-ci serait très basse.

De même, on peut augmenter la précision en donnant très peu de documents en réponse, mais le rappel souffrira.

Il faut donc **utiliser les deux métriques ensemble**.

Les mesures de précision-rappel ne sont pas statiques non plus (c'est-à-dire qu'un système n'a pas qu'une mesure de précision et de rappel). Le comportement d'un système peut varier en faveur de précision ou en faveur de rappel (en détriment de l'autre métrique).

Ainsi, pour un système, on a une courbe de précision-rappel qui a en général la forme suivante:

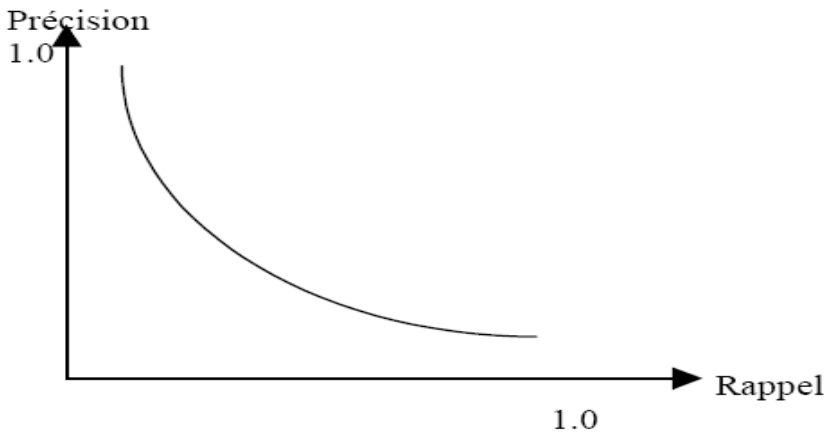


Figure II.10 : courbe de précision-rappel

5.3. Comparaison de systèmes et Précision moyenne

Si on veut comparer deux systèmes de RI, il faut les tester avec le même corpus de test (ou plusieurs corpus de test).

Un système dont la courbe dépasse (c'est-à-dire qu'elle se situe en haut à droite de) celle d'un autre est considéré comme un meilleur système.

Il arrive parfois que les deux courbes se croisent. Dans ce cas, il est difficile de dire quel système est meilleur.

Pour résoudre ce problème, on utilise aussi la *précision moyenne* comme une mesure de performance.

- **La précision moyenne :** est une moyenne de précision sur un ensemble de points de rappel.

On utilise soit la précision moyenne sur 10 points de rappel
(0.1, …, 1.0), ou celle sur 11 points de rappel (0.0, 0.1, …, 1.0).

Cette dernière (11 point de rappel) est possible seulement avec la polarisation.

La **précision moyenne** décrit bien la performance d'un système. C'est la mesure souvent utilisée en RI.

Pour comparer deux systèmes ou deux méthodes, on utilise souvent l'amélioration relative qui est calculée comme suit :

Amélioration de méthode 2 sur méthode 1 = (performance de méthode 2 – performance de méthode 1) / performance de méthode 1.

6. Conclusion

Dans ce chapitre nous avons présenté les approches possibles pour réaliser un système de SRI. Nous avons décrit le processus d'analyse et d'indexation des documents et des requêtes ainsi que la méthode d'évaluation des performances des systèmes de recherche d'information.

Chapitre III :

Réalisation

1 Introduction :

Dans ce chapitre nous allons présenter la structure générale, la modélisation et l'implémentation de notre système pour l'annotation des corpus textuels.

Pour réaliser notre application nous avons utilisé le langage de programmation DELPHI 7. .

2 Objectif Générale du projet :

L'objectif de ce projet est donc :

(i) Construire un corpus annotés de documents pertinent ou non pertinent par rapport à l'utilisateur, le résultat se traduit par un fichier format **XML** ; et un fichier format texte « **.TXT »** ;

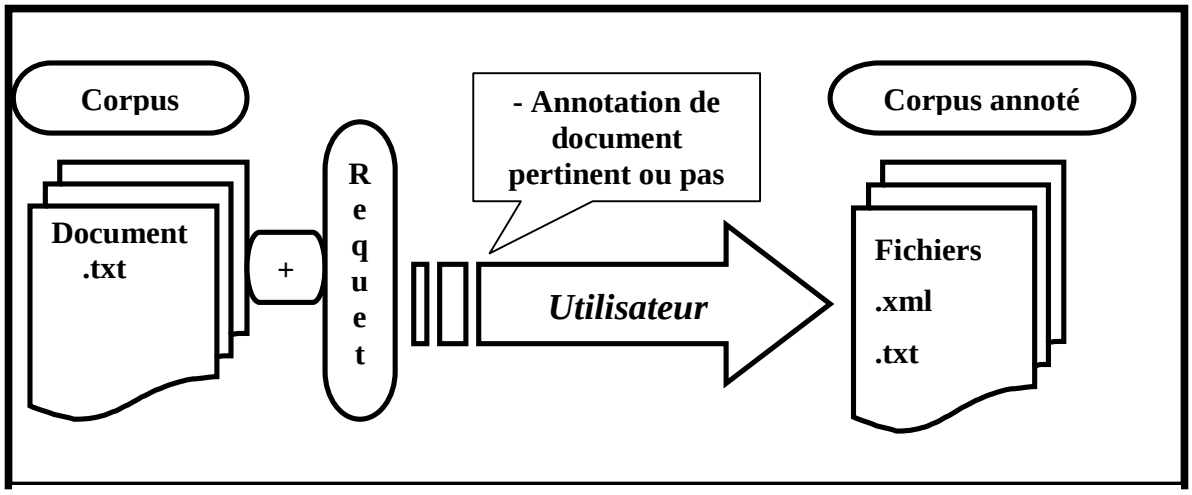


Figure III- 11 : ***Architecture général de l'application***

3 Le langage de programmation :

Pour réaliser notre application nous avons choisi le langage de programmation Delphi 7.

3.1 Delphi 7 :

Delphi ressemble plus à un atelier où l'on dispose d'une boîte à outils et d'un ensemble d'objets qui servent à fabriquer une application. Sous Delphi, on n'écrit pas une application mais on la fabrique.

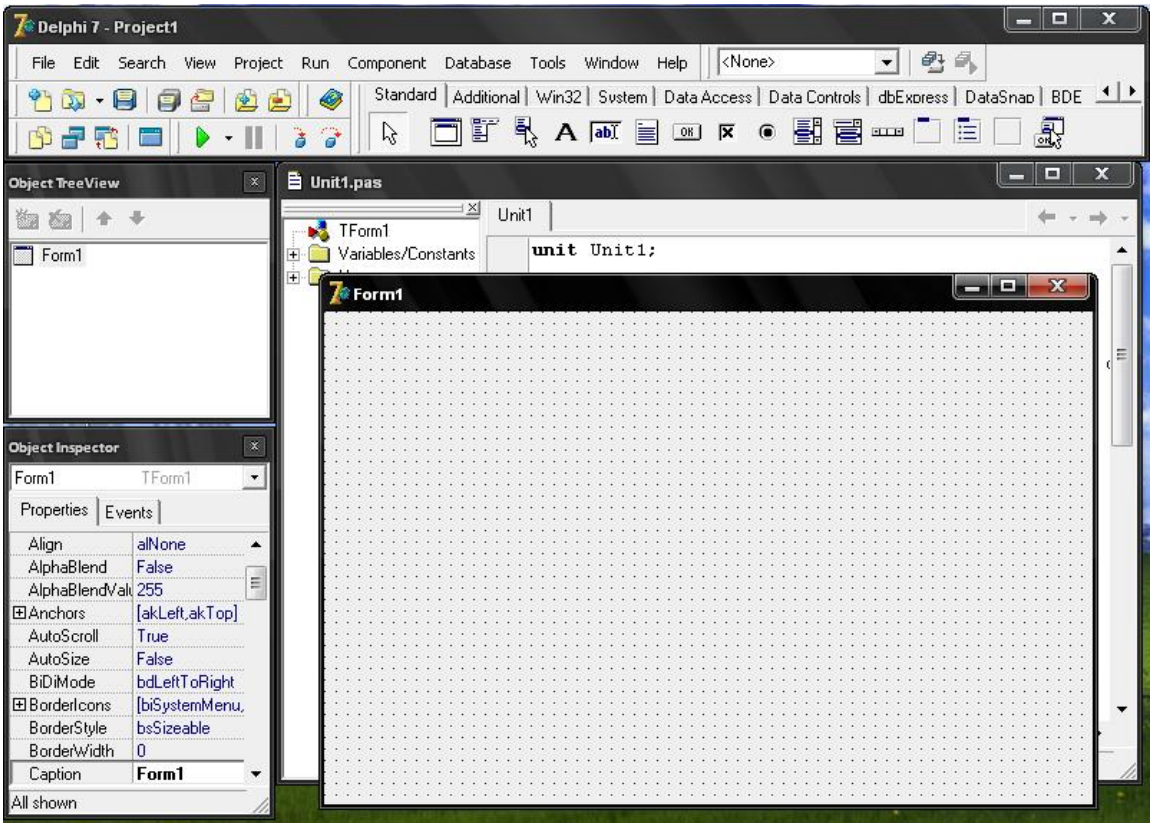


Figure III- 12 : l'interface utilisateur de DELPHI 7

3.2 Le module de base de données de Delphi :

Il permet de faciliter la création des tableaux de données qui vont constituer la base de données. Avant de mettre des données, il est nécessaire d'indiquer la structure de la table c'est à dire les *champs*, le *type* de chaque champ, ainsi que sa *taille*. La BDD dans notre application contient quatre tables : une pour gérer les Requête, une pour gérer les Documents, une pour gérer les Corpus et la dernière pour l'annotation.

4 Modélisation de projet :

4.1 Modèle Conceptuel des Donnée (MCD) :

Le modèle conceptuel des données donne une représentation de l'ensemble des données ainsi que les relations entre ces données, il s'agit donc d'une représentation statique de la réalité.

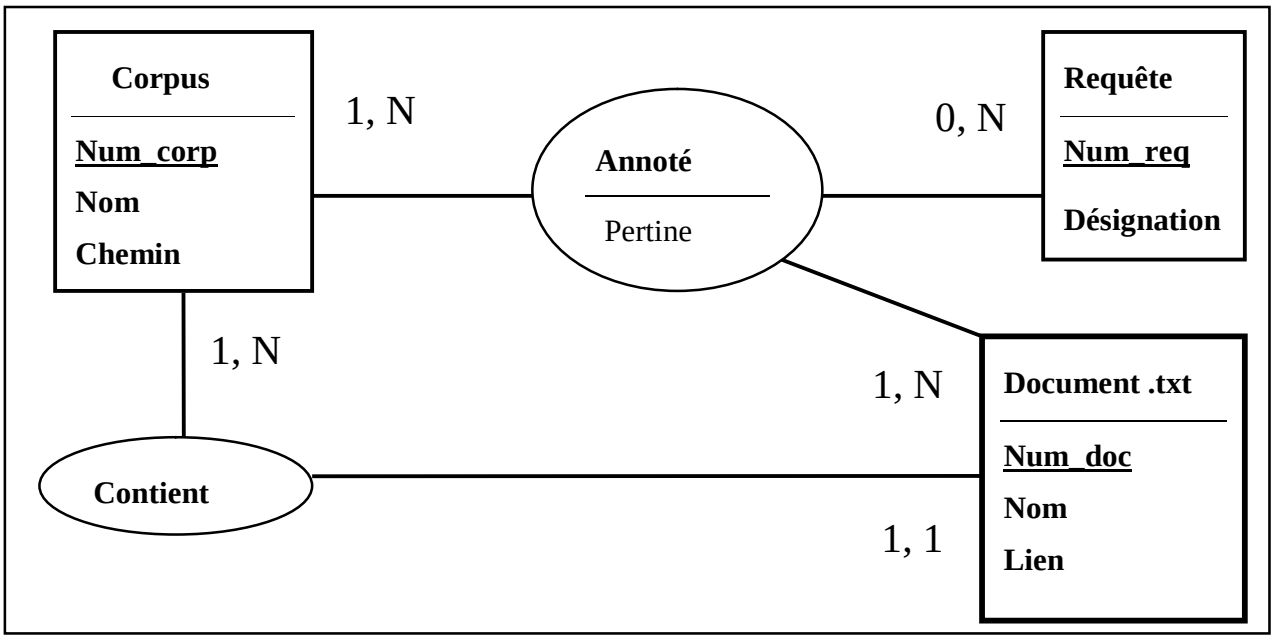


Figure III- 13: Le Modèle Conceptuel des Donnée de notre système

Figure - : Fonctionnement générale de l'application * Le MCD Relationnel de notre

4.2 Modèle Logique des Donnée (MLD) :

Le MLD est une représentation du modèle conceptuel en fonction des possibilités technologiques des matériels et logiciels.

Passage de MCD au MLD :

- **Règle pour les objets du MCD :**
  - L'objet se transforme en table.
  - L'identifiant de l'objet devient la clé primaire de la table.
  - Les propriétés des objets deviennent les champs de la table.
- **Règle pour la relation :**
  - **Cas des relations de type « père, fils » :**

Cardinalité objet de type « père» (0, n) ou (1, n).

Cardinalité objet de type « fils» (0,1) ou (1,1).

L'objet père devient table père.

L'objet fils devient table fils.

L'identifiant de l'objet père devient attribut de l'objet fils. Ce dernier appelé clé étrangère.

Les propriétés de la relation deviennent des attributs de la table fils.

  - **Cas des relations de type « non père, fils » :**

Cardinalité objet (0, n) ou (1, n).

Un objet détermine une table, l'identifiant de l'objet devient la clé de la table.

Une relation devient une table l'identifiant de la relation devient la clé primaire de la table.

Le MLD Relationnel de notre application :

- **Requête** (Num_req , Désignation).
- **Document** (Num_doc, Nom, Lien, Num_corpus).
- **Corpus** (Num_corpus, Nom, Chemin).
- **Annoter** (Num_req, Num_doc, Num_corpus, Pertinent).

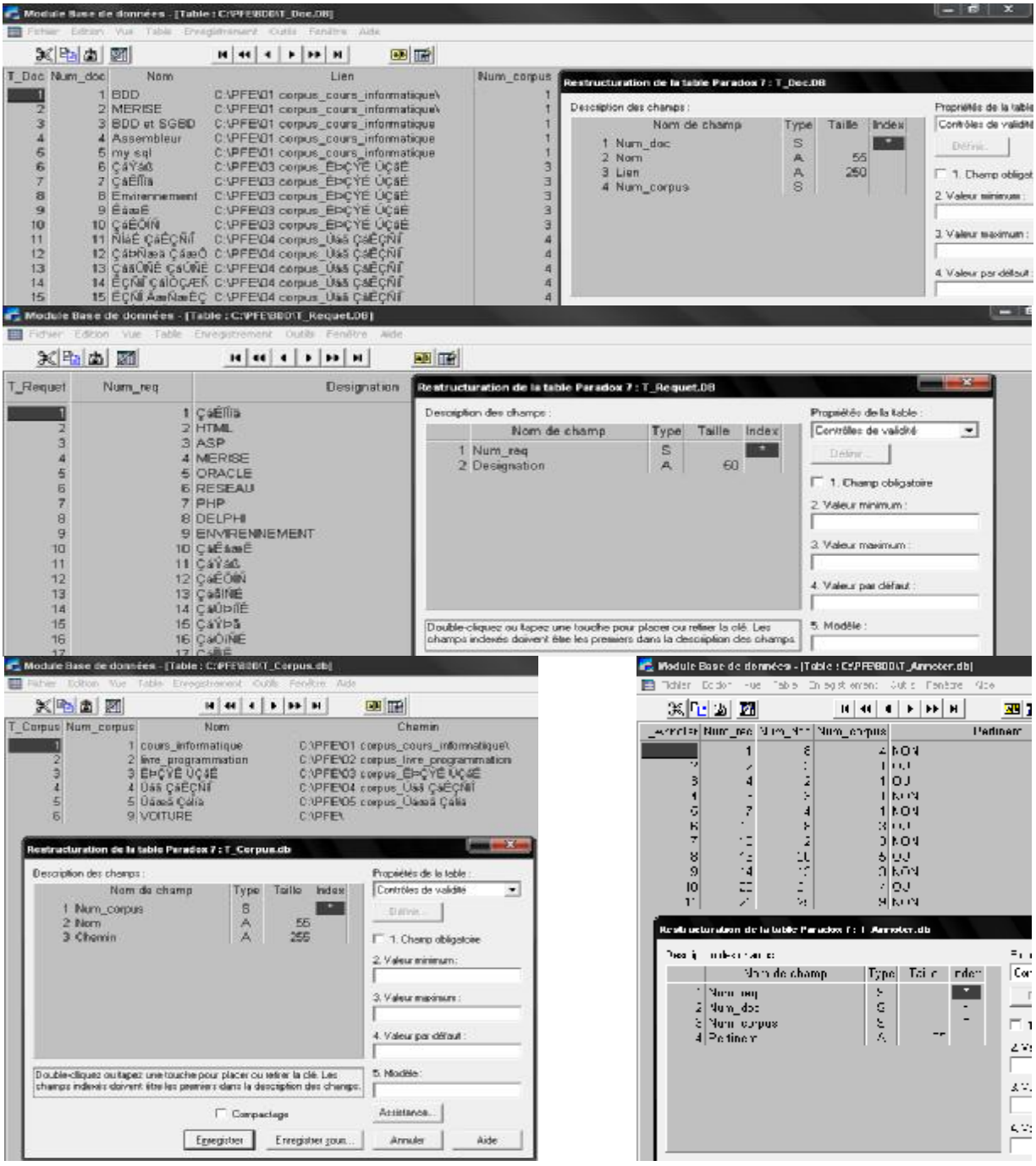
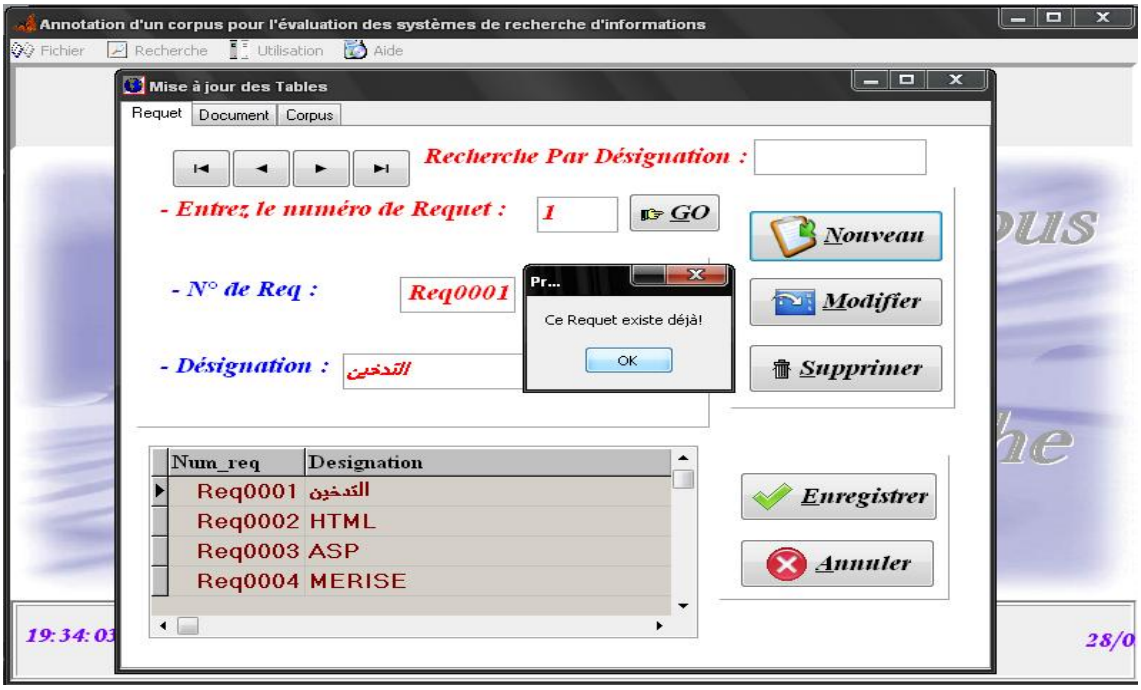


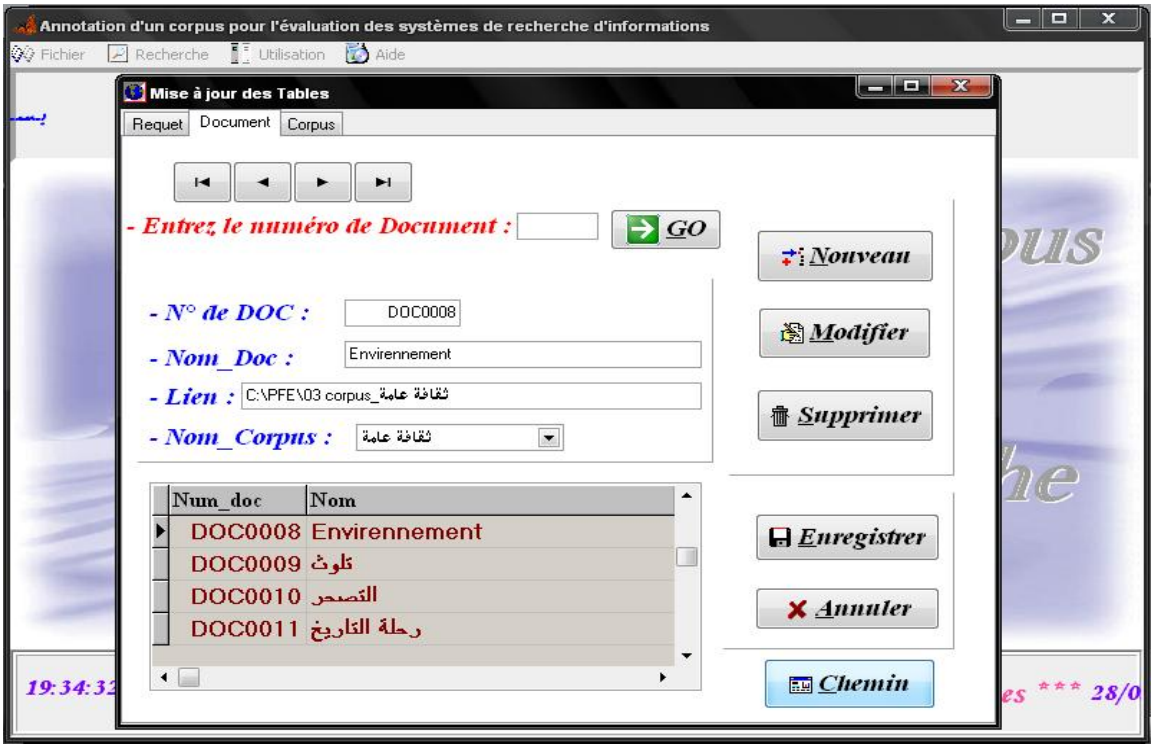
Figure III- 14 : Création de la BDD de notre application (différentes tables)

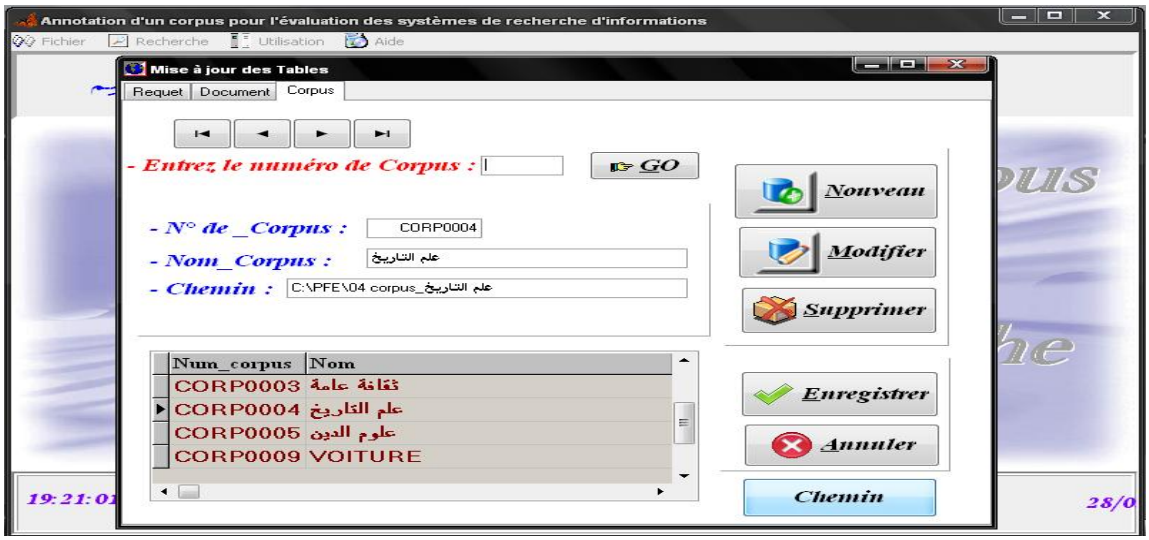
5 - Description de l'application :

Figure III-15 :l'interface principale de l'application

⟶ : Comme première étape remplir les tables avec la possibilité de faire la mise à jour (Modifier, Supprimer, consulter)


Annotation d'un corpus pour l'évaluation des systèmes de recherche d'informations
Fichier
Recherche
Utilisation
Aide
Mise à jour des Tables
Requet
Document
Corpus
Recherche Par Désignation :
- Entrez le numéro de Requet :
1
GO
Nouveau
- N° de Req :
Req0001
Modifier
Pr...
Ce Requet existe déjà!
OK
- Désignation :
التدخين
Supprimer
Num_req
Designation
Req0001 التدخين
Req0002 HTML
Req0003 ASP
Req0004 MERISE
Enregistrer
Annuler
19:34:03
28/0


Annotation d'un corpus pour l'évaluation des systèmes de recherche d'informations
Fichier
Recherche
Utilisation
Aide
Mise à jour des Tables
Requet
Document
Corpus
- Entrez le numéro de Document :
GO
Nouveau
- N° de DOC :
DOC0008
Modifier
- Nom_Doc :
Envirennement
- Lien :
C:\PFE\03 corpus_ثقافة عامة
Supprimer
- Nom_Corpus :
ثقافة عامة
Num_doc
Nom
DOC0008 Envirennement
DOC0009 تلوث
DOC0010 التصحر
DOC0011 رحلة التاريخ
Enregistrer
Annuler
Chemin
19:34:32
28/0



: La deuxième étape : l'utilisateur spécifie le corpus et le fichier textuel à annoté avec l'information pertinent ou non pertinent en relation à une requête.

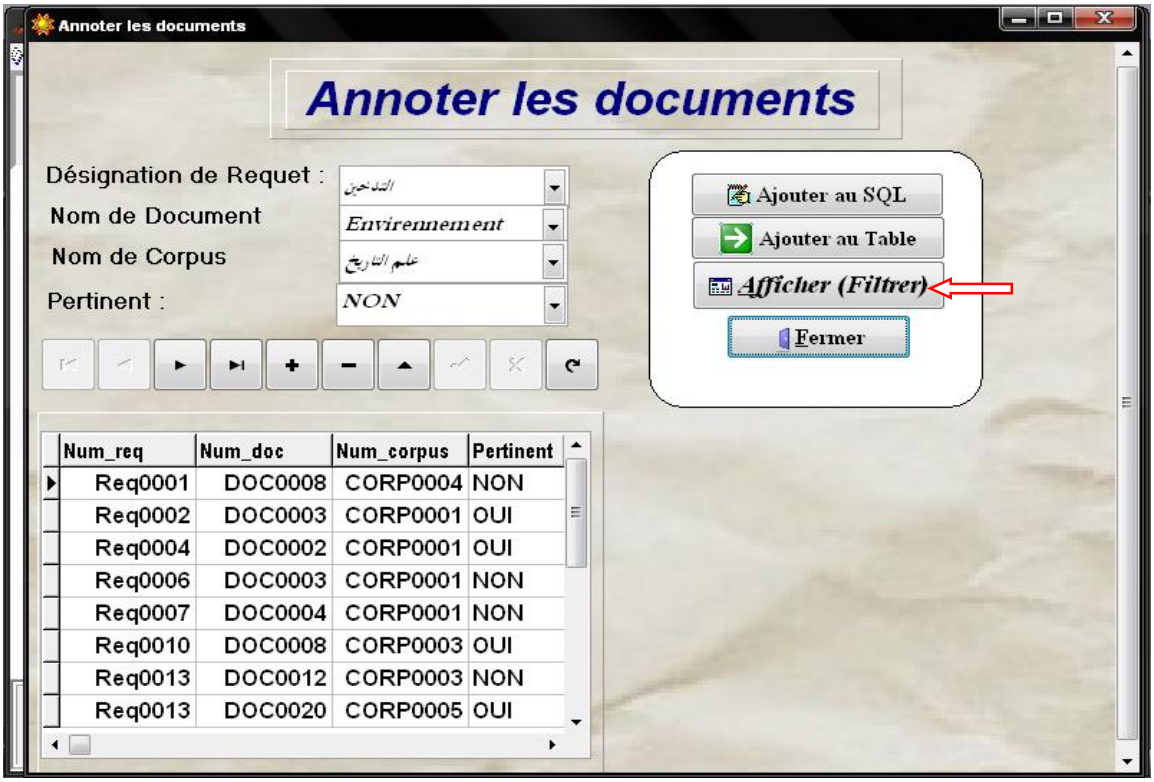
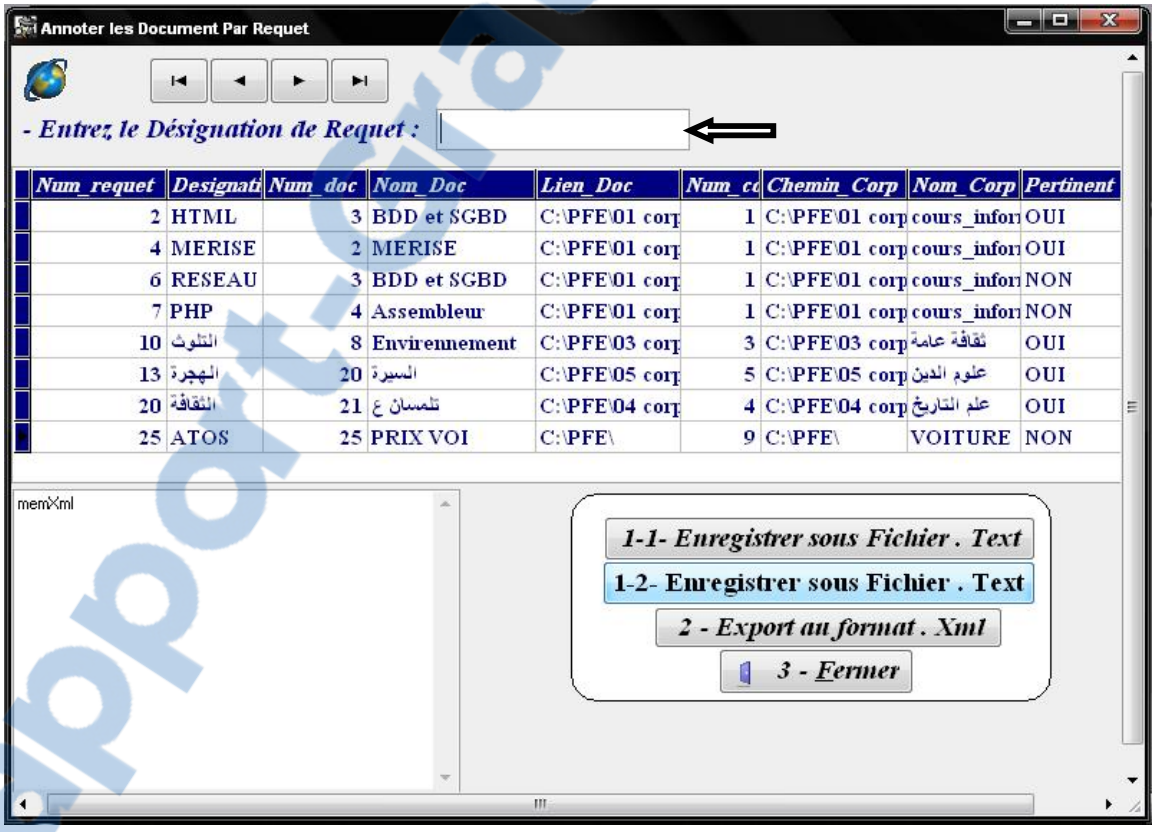


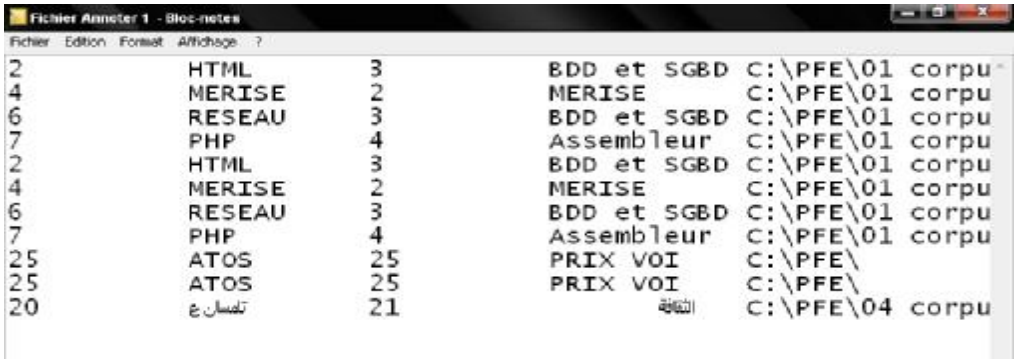
Figure III-16 : Chargement du fichier pour annotation

⟹ : La troisième étape c'est la visualisation : si on clique sur le bouton ***Afficher*** un tableau résumant plusieurs informations est construit. Ce tableau est développé par le composant ***Query*** avec la requête SQL suivant :

```
SELECT DISTINCT d.Num_req, d.Num_doc, d.Num_corpus, d.Pertinent, d1.Num_corpus, d1.Nom,
d1.Chemin, D2.Num_doc, D2.Nom, D2.Lien, D2.Num_corpus, D3.Num_req, D3.Designation
FROM "C:\PFE\BDD\T_Annoter.db" d, "C:\PFE\BDD\T_Corpus.db" d1, "C:\PFE\BDD\T_Doc.DB" D2,
"C:\PFE\BDD\T_Requet.DB" D3
WHERE
(d1.Num_corpus = d.Num_corpus)
 AND (D2.Num_doc = d.Num_doc)
 AND (D2.Num_corpus = d1.Num_corpus)
 AND (D3.Num_req = d.Num_req)
ORDER BY d.Num_req, d.Num_doc, d.Num_corpus, d.Pertinent, d1.Num_corpus, d1.Nom, d1.Chemin,
D2.Num_doc, D2.Nom, D2.Lien, D2.Num_corpus, D3.Num_req, D3.Designation
```



- On clique sur le bouton ***1-1- Enregistrer sous Fichier. Txt pour générer un résultat sous la forme d'un fichier textuel.*** La **figure 17** montre son contenu.



- On clique sur le bouton ***1-2- Enregistrer sous Fichier. Txt*** nous avons présentez le figure suivant :

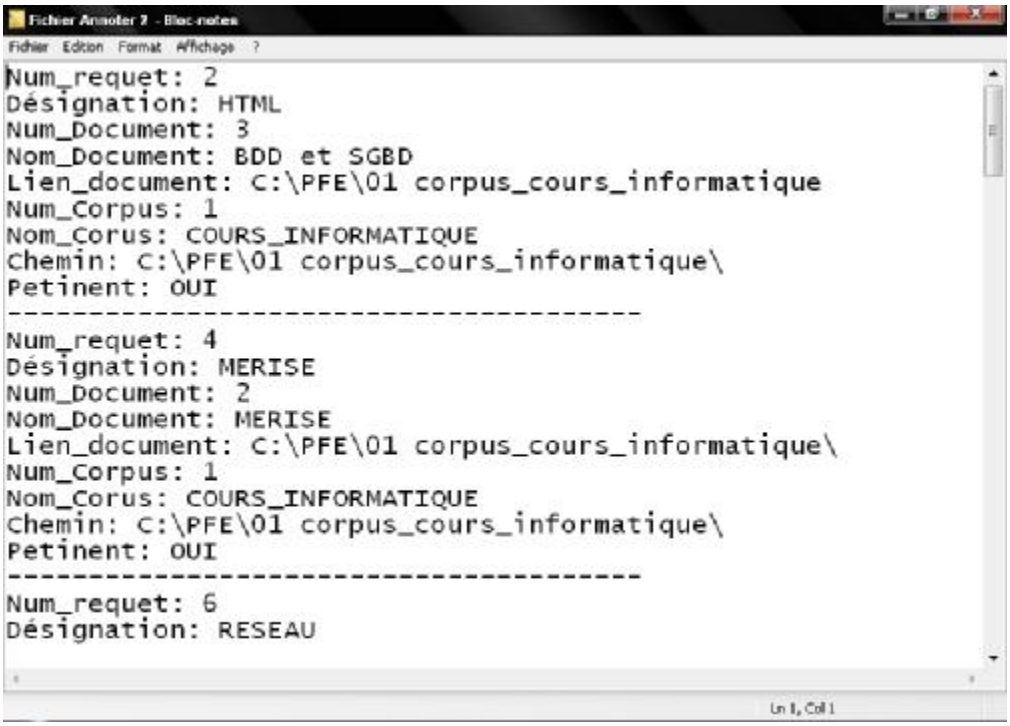


Figure III-17 : Visualisation de l'annotation d'un document .txt

- On clique sur le bouton ***2 - Export au format. XML*** pour exporter le résultat sous format XML comme le montre la ***figure 18***.

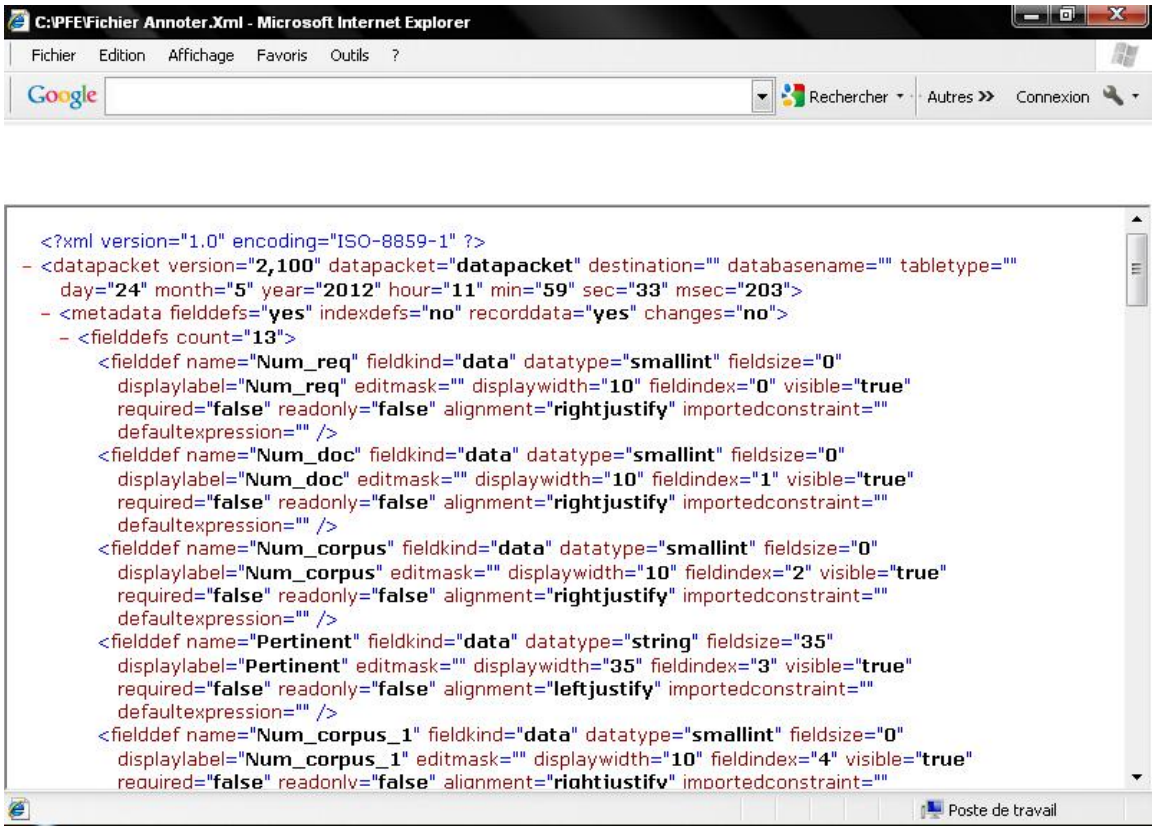
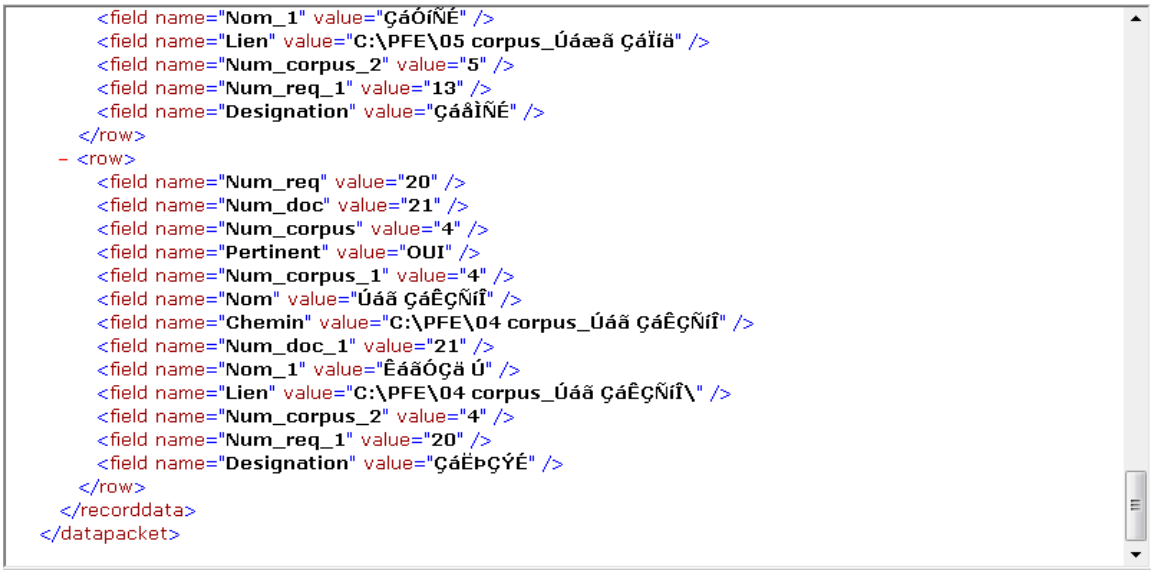
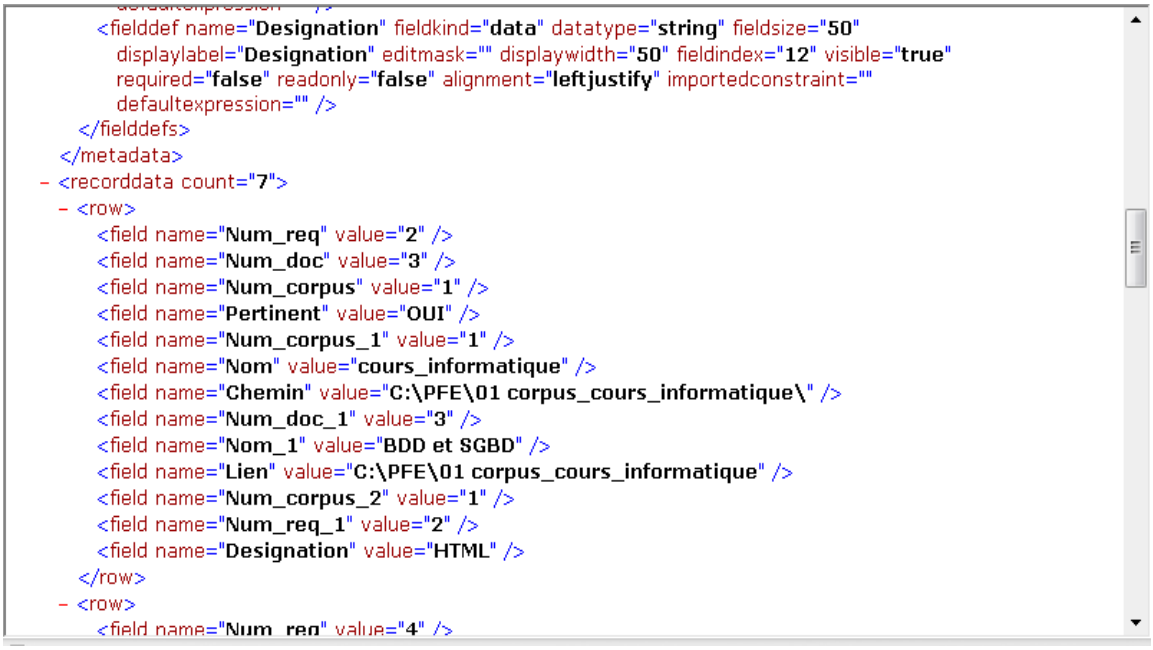


Figure III - 18 : Format XML d'une annotation





```
      <fielddef name="Designation" fieldkind="data" datatype="string" fieldsize="50"
        displaylabel="Designation" editmask="" displaywidth="50" fieldindex="12" visible="true"
        required="false" readonly="false" alignment="leftjustify" importedconstraint=""
        defaultexpression="" />
    </fielddefs>
  </metadata>
- <recorddata count="7">
  - <row>
      <field name="Num_req" value="2" />
      <field name="Num_doc" value="3" />
      <field name="Num_corpus" value="1" />
      <field name="Pertinent" value="OUI" />
      <field name="Num_corpus_1" value="1" />
      <field name="Nom" value="cours_informatique" />
      <field name="Chemin" value="C:\PFE\01 corpus_cours_informatique\" />
      <field name="Num_doc_1" value="3" />
      <field name="Nom_1" value="BDD et SGBD" />
      <field name="Lien" value="C:\PFE\01 corpus_cours_informatique" />
      <field name="Num_corpus_2" value="1" />
      <field name="Num_req_1" value="2" />
      <field name="Designation" value="HTML" />
    </row>
  - <row>
      <field name="Num_req" value="4" />
```

6 - Conclusion :

Dans ce chapitre nous avons présenté la partie modélisation et implémentation de notre application. Il reste maintenant la partie utilisation pour instancier la base de données puis l'utilisation de notre application dans un cas réel d'annotation.

Conclusion générale

Conclusion générale

Suite aux grandes quantités des documents,

Les travaux présents dans ce mémoire se situent dans le cadre de la conception et la réalisation d'un outil pour l'évaluation d'un système de recherche d'informations ; notre tâche principale consiste à réaliser un logiciel pour l'annotation des corpus.

Le but des systèmes de recherche d'information est de récupérer des documents pertinents dans différentes langues répondants a un besoin utilisateur exprime dans une langue différente de celles des documents.

En termes de ce mémoire un outil pour l'annotation est réalisé, reste maintenant son utilisation dans un cas réel pour l'évaluation d'un SRI. Les perspectives de ce travail sont nombreuses, nous pouvons citer à titre d'exemple l'ajout des modules pour le calcul des différents paramètres permettant l'évaluation des SRI comme la précision et le rappel. Une autre perspective concerne l'ajout d'un autre module pour la visualisation et donc la comparaison des systèmes évalués.

.

Bibliographies

Bibliographies

[NawelNassr01], Croisement de langues en recherche d'information : traduction et désambiguïsation de requêtes, THÈSE de Doctorat de l'Université Paul Sabatier V 2002

[Abderrahim M,A 04] Recherche d'information/Analyse et Indexation 2010/2011

[A .Rozenknop 04], Master MICR Paris 13 , Recherche et extraction d'information Source : Romaric Besancon CEA-LIST/LIC2M

[Bla90] DC. Blair. Language and representation in information retrieval.In Proceedings of Elesevier Science Publications, Amsterdam, 1990.

[Salton 1970] G. Salton, The SMART retrieval system: Experiments in automatic documentprocessing. Prentic pages : 50-57, 1970.

[Karbasi, 07] S. Karbasi : Pondération des termes en Recherche d'Information: Modèle de pondération basé sur le rang des termes dans les documents. Thèse de doctorat, université Paul Sabatier (2007).

[Kiryakov, 04] Kiryakov A, Popov B, Terziev I., Manov D., Ognyanoff D, Semantic Annotation, indexing, and retrieval, Journal of Web Semantics, 2(1), pp49-79, (2004).

[Ponte & al, 98] Ponte, J. and Croft, W, A language modelling approach to information retrieval. In Proceedings of the 21st Annual International ACM SIGIR

[Boughanem et al 1991] M. Boughanem, J. Mothe, C. Soule-Dupuy, Optimizationof knowledge representation in information retrieval. Proceedingsof NEURO-NIMES'91, pages : 111-117, 1991.

[Boughanem 1992] M. Boughanem Systèmes de recherche d'information d'unmodèleclassique à un modèle connexioniste Thèse de l'université Paul.Sabatier de Toulouse. 1992.

[Boughanem et al 1994] M. Boughanem, C. Soule-Dupuy, Query expansionand neural networks. In Proceedings of RIAO '94, pages : 124-131,1994.

[Boughanem 1998] M. Boughanem, T. Dkaki, J. Mothe& C. SouleDupuy, Mercure at TREC-7. Proceedings of TREC-7, Pages : 135-141,1998.

[Boughanem et al 1999a] M. Boughanem, C. Chrisment, L. Tamine, Query space exploration based on genetic algorithms. Information Retrieval Journal. pages : 125-136, 1999

[Boughanem 2000] M. Boughanem, Formalisation et spécification des systèmes de recherche et de filtrage d'information, HDR de l'université Paul Sabatier de Toulouse.