THESE

présentée devant

L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

pour 1' obtention du

doctorat troisième cycle

SPECIALITE INFORMATIQUE

par

JEAN-MARIE DAVID

Service Commun de la Documentation
INPL
Nancy-Brabois

SUJET : SIMULATION DE L'ACTIVITÉ D'UN DOCUMENTALISTE

80-T-056

soutenue le 3 décembre 1980 devant la commission composée de

D 136 036415 7

Président : M. C.PAIR

Examinateurs : MM. M.BORILLO
D.COULON
D.KAYSER
J.M.PIERREL

136036 4157

(FI) 1980 DAVAINE, J.-M.

THESE

présentée devant

L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

pour l' obtention du

doctorat troisième cycle

SPECIALITE INFORMATIQUE

par

JEAN-MARIE DAVID

SUJET : SIMULATION DE L'ACTIVITÉ D'UN DOCUMENTALISTE

Service Commun de la Documentation
INPL
Nancy-Brabois

80-T-056

BIBLIOTHÈQUE - ÉCOLE DES MINES
PARC DE SAURUPT - 54042 NANCY

soutenue le 3 décembre 1980 devant la commission composée de

Président : M. C.PAIR
Examinateurs : MM. M.BORILLO
D.COULON
D.KAYSER
J.M.PIERREL

13/109

J'adresse mes plus vifs remerciements

à mon épouse qui, par son soutien moral et son aide, m'a permis de me consacrer dans de bonnes conditions à cette thèse

à Monsieur PAIR de l'honneur qu'il me fait d'avoir accepté la présidence du jury

à Monsieur COULON qui m'a guidé avec beaucoup d'attention et de compétence

à Monsieur KAYSER et Monsieur PIERREL pour leurs critiques constructives

à Monsieur BORILLO qui a bien voulu faire partie du jury

à Madame BEAURAIN pour la réalisation matérielle de cette thèse

J'adresse mes plus vifs remerciements

à mon épouse qui, par son soutien moral et son aide, m'a permis de me consacrer dans de bonnes conditions à cette thèse

à Monsieur PAIR de l'honneur qu'il me fait d'avoir accepté la présidence du jury

à Monsieur COULON qui m'a guidé avec beaucoup d'attention et de compétence

à Monsieur KAYSER et Monsieur PIERREL pour leurs critiques constructives

à Monsieur BORILLO qui a bien voulu faire partie du jury

à Madame BEAURAIN pour la réalisation matérielle de cette thèse

- S O M M A I R E -

I. INTRODUCTION : NOS OBJECTIFS 6

II. PRÉSENTATION DE L'ÉTUDE 12

1. Introduction 13

2. Analyse des systèmes documentaires actuels 13

2.1. Les inconvénients de la classification décimale universelle 13

2.2. L'automatisation de la gestion du prêt 14

2.3. L'automatisation des recherches documentaires 15

2.3.1. L'utilisation d'un langage documentaire 15

2.3.2. L'utilisation d'un module de traduction langage naturel → langage documentaire 22

3. Présentation de l'étude 26

3.1. Ce que nous avons fait 26

3.2. Ce qui pourrait être fait 33

III. L'ANALYSE AUTOMATIQUE DES REQUETES 35

1. Le filtrage des descripteurs d'une requête 36

2. L'importance des différents objectifs d'une requête 38

2.1. Influence d'un ordre de priorité sur la stratégie de la recherche documentaire 39

2.2. Etablissement d'un ordre de priorité entre les objectifs d'une requête grâce à son analyse syntaxique 40

2.2.1. Définition de l'analyse syntaxique 40

2.2.2. But de l'analyse syntaxique 40

2.2.3. Principe d'analyse syntaxique 41

2.2.4. Règles d'analyse syntaxique 42
2.2.5. Exemples d'analyse syntaxique 47
2.2.6. Règles de priorité des objectifs d'une requête 49
2.2.7. Nécessité et limites d'un analyseur syntaxique 50
3. L'amplification de la portée des requêtes 53
3.1. Particularités de l'amplification des requêtes présentées à notre système 54
3.2. L'amplification de la portée d'une requête grâce au thésaurus 55
3.2.1. Le thésaurus 55
3.2.1.1. Rôle 55
3.2.1.2. Description des liens sémantiques 55
3.2.1.2.1. Les relations 56
3.2.1.2.2. Les plausibilités de proximité 58
3.2.1.2.2.1. Composition des plausibilités de proximité 59
3.2.1.2.2.2. Les plausibilités relationnelles 61
3.2.1.2.2.3. Les plausibilités d'intérêt 62
3.2.1.2.2.4. Calcul de la plausibilité de proximité de deux objectifs en relation directe 63
3.2.1.2.3. Les domaines de validité 64
3.2.1.3. Exemples de liens sémantiques dans un thésaurus 66

3.2.1.4. Mode de construction du thésaurus 67
3.2.2. Description de l'amplification de la portée d'une requête grâce à un thésaurus 70
3.2.2.1. Principe général 70
3.2.2.2. Composition des relations 71
3.2.2.3. Calcul de la proximité de deux objectifs reliés indirectement 74
3.2.2.4. Test de la similarité d'un domaine de validité par rapport à une requête 76
3.2.3. Conclusion : les limites d'un thésaurus 81
3.3. L'amplification de la portée d'une requête grâce aux procédures stratégiques 82
3.3.1. Définition et rôle des procédures stratégiques 82
3.3.2. Description et exploration des procédures stratégiques 84
3.4. Conclusion 93

IV. LA MODÉLISATION DU FONDS DOCUMENTAIRE 94
1. Description et exploration des procédures documentaires 95
2. Mode de construction des procédures documentaires et stratégiques 103

V. LA RECHERCHE DOCUMENTAIRE 105
1. Introduction 106
2. Particularités de la recherche documentaire effectuée par notre système 107
3. Les sorties désirées 110
4. Proximité d'un objectif par rapport à un chemin 111

5. Proximité de deux chemins 113
5.1. Définition d'un chemin idéal 113
5.2. Plausibilité d'atténuation 114
5.3. Détermination de la plausibilité de proximité de deux chemins 116
6. Conditions d'arrêt d'une recherche documentaire 117
7. Stratégie de la recherche documentaire 120
7.1. Principe de la stratégie de la recherche documentaire 120
7.2. Ordre de priorité des questions 125
7.3. Importance des différentes questions 127
7.4. Etude de la stratégie sur un exemple 130
8. Les procédures structurales 139
8.1. Définition 139
8.2. Liste des procédures structurales 139
8.3. Exemple de détermination de réponses à des questions qui appellent des procédures structurales 142
9. La procédure spécialité 146
10. La procédure message 146
11. Conclusion 147

VI. PROGRAMMATION 148
1. Introduction 149
1.1. Choix du langage de programmation 149
1.2. Les étapes de la programmation 150
2. La création des connaissances 151
2.1. La création du lexique 151

2.1.1. Représentation interne 151
2.1.2. Organisation du lexique 151
2.1.2.1. Détermination de la classe d'un objectif 153
2.1.2.2. Calcul des adresses de début et de fin des classes 153
2.2. La création du thésaurus 155
2.2.1. Contrôles sur la validité des liens sémantiques 156
2.2.2. Subdivision du thésaurus en classes 157
2.3. La création des procédures stratégiques 160
2.3.1. Transcription sur cartes 160
2.3.2. Contrôle et rangement des procédures stratégiques 161
2.3.3. Edition des procédures stratégiques 162
2.4. La création des procédures documentaires 163
3. La recherche documentaire 164
3.1. Les sorties désirées 164
3.2. Les entrées du module "recherche documentaire" 164
3.3. Contrôles sur la validité des requêtes 166
3.4. La stratégie de recherche 166
4. Conclusion 166

VII. CONCLUSION GÉNÉRALE 167

ANNEXES 172

GLOSSAIRE

BIBLIOGRAPHIE

I. INTRODUCTION :

NOS OBJECTIFS

I. INTRODUCTION

Imaginons l'activité d'un documentaliste : il voit arriver un lecteur qui s'adresse à lui. Le lecteur exprime ses besoins dans une requête dont la précision peut être très variable:

Il peut être demandé, par exemple, le livre intitulé "Le système documentaire Satin 1, écrit par E.CHOURAQUI..." . Dans ce cas, les caractéristiques formelles des documents (titre, auteur, éditeur, année d'édition, lieu d'édition, numéro de tome...) permettent de sélectionner le livre demandé, si ce dernier appartient au fonds documentaire à explorer.

Mais, le plus souvent, les lecteurs ne savent pas exactement les livres qu'ils désirent : leur requête porte sur un certain sujet. (exemple : "je voudrais des livres se rapportant à l'automatisation des recherches documentaires"). A partir de la connaissance du contenu des documents, le documentaliste doit alors rechercher un certain nombre de livres qui sont susceptibles d'intéresser le lecteur. C'est à ce deuxième type de requête que nous allons nous intéresser dans cette thèse.

L'accroissement rapide du nombre des documents et des lecteurs rend la tâche d'un documentaliste de plus en plus difficile et inefficace. C'est la raison principale pour laquelle on cherche à automatiser cette tâche grâce à la construction de systèmes documentaires qui, à partir d'informations documentaires enregistrées en mémoire, auraient pour rôle de fournir aux utilisateurs présentant une requête, les références des livres qui paraissent convenir le mieux.

Imaginons à présent un lecteur, lorsqu'il va s'adresser à un système documentaire : ses besoins, dès lors, ne seront plus appréhendés qu'à partir des signes graphiques.

La langue écrite se trouve privée de ressources dont dispose la langue parlée : l'allure du débit, les inflexions, le timbre de la voix, les silences, la mimique...

Les besoins souvent imprécis du lecteur ne seront donc traduits que par une requête écrite qui, fréquemment, sera courte et qui, la plupart du temps, ne reflètera qu'imparfaitement les désirs de l'utilisateur.

En discutant avec les lecteurs, un documentaliste disposait d'atouts supplémentaires : l'impression donnée par un lecteur et la manière avec laquelle il exprimait ses besoins influençaient en effet le documentaliste dans sa recherche documentaire.

De plus, nous verrons (chapitre II) qu'une requête et un document doivent être généralement représentés par quelques mots-clés pour que le système documentaire puisse leur donner une signification et effectuer une recherche documentaire.

Afin de lutter contre ces pertes d'informations que provoque généralement l'automatisation de la tâche d'un documentaliste, nous avons orienté notre étude vers l'intégration et l'utilisation des connaissances d'un documentaliste dans un système documentaire.

L'objectif n'est pas seulement pratique : nous voulons aussi tester une certaine conception de la représentation des connaissances : il s'agit là d'un problème important en Intelligence Artificielle.

La documentation automatique nous semble être un cadre adéquat pour effectuer ce test, pour plusieurs raisons : les principaux problèmes qui se posent en documentation automatique -modélisation de documents, compréhension de la requête d'un utilisateur, élaboration de réponses en confrontant la requête à la description des documents- paraissent avoir un rapport étroit avec le problème de la représentation des connaissances. Ceci implique que l'efficacité des méthodes de recherches documentaires va être étroitement liée aux moyens dont on va disposer pour représenter l'information.

D'autre part, pour simuler l'activité d'un documentaliste, il semble inévitable de répondre aux questions suivantes :

- quelles sont les connaissances d'un documentaliste qui paraissent utiles pour cour onner de succès les recherches documentaires ?
- comment les intégrer dans un système documentaire ? (en particulier, comment les structurer?)
- à partir du modèle de réprésentation défini, comment explorer judicieusement ces connaissances ? (pour que les requêtes puissent déclencher des recherches conformément à des stratégies pré-définies).

Ces questions (en particulier la 1ère et la 3ème) n'admettent certainement pas une réponse unique : les documentalistes ont une façon plus ou moins personnelle de procéder pour atteindre

les documents pertinents. Les démarches que nous proposons introduiront également une certaine part d'intuition et de subjectivité difficilement évitable.

Les connaissances que nous apporterons au système se rapporteront en particulier aux lecteurs, à la description des approximations et à l'analyse du langage naturel : nous pensons que ces informations sont en effet très précieuses pour appréhender convenablement les désirs des utilisateurs, conduire certaines formes d'inférences et orienter la recherche documentaire vers les documents qui correspondent le mieux à leurs besoins.

La nature des informations à intégrer dans notre système documentaire implique que nous désirons simuler l'activité d'un documentaliste qui appartiendrait à une bibliothèque dont la taille serait plutôt petite.

Après une présentation générale de notre système documentaire dans laquelle nous essayerons de mettre en évidence l'originalité de notre étude, nous montrerons comment sont analysées les requêtes, puis nous décrirons la représentation du fonds documentaire avant d'indiquer les caractéristiques principales de la recherche documentaire.

Après la conception du système, nous montrerons brièvement comment nous l'avons réalisé en décrivant la programmation de la représentation de ses connaissances et de ses recherches documentaires.

Nous illustrerons les traitements documentaires en considérant des requêtes qui porteront essentiellement sur l'informatique.

II. PRESENTATION

DE L'ETUDE

II. PRESENTATION DE L'ETUDE

1. INTRODUCTION

L'informatique documentaire est un domaine dans lequel de nombreuses recherches ont déjà été effectuées. Pour situer notre travail, nous allons rappeler sommairement les tentatives qui ont été effectuées et nous discuterons de façon plus approfondie ce qui concerne la représentation des requêtes et des documents.

2 . ANALYSE DES SYSTÈMES DOCUMENTAIRES ACTUELS

2.1. Les inconvénients de la Classification Décimale Universelle (CDU)

Les livres sont classés généralement par matière pour qu'un documentaliste puisse les retrouver rapidement et facilement. Il n'est pas prévu d'autres modes d'accès dans la CDU [23](1).

---

(1) Par convention, les indices mis entre parenthèses dans le texte de notre thèse renvoient à des notes écrites au bas de la même page que celle où figurent ces indices.
Un nombre i mis entre crochets : [i] renvoit au ième document de la liste des références bibliographiques (dernières pages).

On assiste hélas à une explosion de l'information documentaire si bien que les documentalistes, qui ne peuvent naturellement pas être spécialistes dans tous les domaines, ne peuvent plus s'orienter dans la CDU qui, en 1973, comptait déjà plus de 150 000 rubriques [34]. La quantité tend à noyer la qualité des services d'un documentaliste.

De plus, toute structuration hiérarchique des connaissances implique des délimitations délicates, des localisations difficiles pour les documents situés près d'une frontière et un rattachement arbitraire de certains concepts à certaines disciplines.

L'informatique permet notamment de multiplier les chemins d'accès à un document.

2.2. L'automatisation de la gestion du prêt

Des systèmes ont d'abord été conçus pour automatiser la gestion du prêt [29]. Il s'agit d'un problème de gestion relativement simple consistant essentiellement à enregistrer les transactions grâce auxquelles il est possible d'intercepter les réservations, d'éditer les lettres de réclamation et d'effectuer des analyses statistiques.

2.3. L'automatisation des recherches documentaires

Puis des systèmes documentaires ont été réalisés : rappelons que leur rôle principal est à peu près similaire à celui des documentalistes : à partir de données documentaires enregistrées en mémoire, ils essayent de fournir aux utilisateurs présentant une requête, les références des livres qui paraissent convenir le mieux.

Les travaux doivent donc porter essentiellement sur :
- la compréhension des requêtes
- la représentation du contenu des documents du fonds documentaire
- la définition d'une fonction de recherche

Nous allons à présent analyser les différentes tentatives qui ont été effectuées pour essayer de résoudre ces problèmes fondamentaux.

2.3.1. L'utilisation d'un langage documentaire

L'un des problèmes fondamentaux de la recherche documentaire est l'extraction d'information à partir de données textuelles en Langage Naturel (L.N.). Au cours des années 60, les premiers travaux de recherche effectués dans ce domaine ont effectivement fait

appel à une analyse linguistique [10g]. Ces travaux n'ont pas eu le succès escompté, que ce soit pour l'indexation automatique ou pour l'analyse de questions : la hardiesse de la tâche et le manque de bons outils rendaient les processus de reconnaissance trop complexes. Le L.N. est souvent ambigu, vague et n'est pas défini par un modèle formel comme un langage de programmation par exemple.

Cette dernière observation accrédite l'idée d'exprimer directement une requête ou un document dans un Langage Documentaire formel (L.D.).

Un L.D. permet la normalisation de l'expression du contenu des textes, afin que les notions ou les sujets analogues soient toujours désignés par les mêmes termes (réduction des cas de synonymie). Réciproquement, il serait souhaitable qu'aucun terme utilisé pour l'analyse ne soit jamais pris dans des sens différents (réduction des cas de polysémie) mais c'est pratiquement impossible car il semble qu'il n'y ait pas de limitation dans le nombre de rôles que peut jouer un terme [6 (1er rapport)].

Grâce à ces termes normalisés, les requêtes et les documents peuvent être représentés par un ensemble de mots-clés non hiérarchisés. La méthodologie employée consiste alors dans un premier temps à reconnaître et extraire les concepts du texte étudié puis à traduire ces concepts en L.D.

Cette façon de procéder paraît simple et provoque une multiplication des chemins d'accès aux livres.
Le principe a été adopté dans Mistral III/IV : le fonds documentaire est alors représenté par un "fichier inversé" indiquant la présence ou l'absence d'un concept dans un document :

	document 1	document 2	document 3	...
concept 1	1	0	0	
concept 2	0	1	0	
concept 3	1	0	1	
⋮				

Le principe du fichier inversé a été également suivi dans Smart [27][28] mais les représentations paraissent plus précises puisqu'elles comportent l'indication de l'importance des concepts dans les requêtes et les documents, évaluée en fonction de leur fréquence d'apparition. Un document ou une requête peuvent alors être exprimés en fonction d'une base vectorielle composée de n vecteurs de base correspondant aux n concepts symbolisés par le vocabulaire du L.D.

Le mode de représentation vectorielle ne fait pas apparaître une hiérarchie entre les mots-clés d'une requête. Au contraire, avec les systèmes Mistral III/IV et Satin, par exemple, un utilisateur donne une priorité à un mot-clé définissant un thème de recherche. La recherche documentaire consiste alors à sélectionner dans une première étape les documents se rapportant au thème. Le résultat est sauvegardé. Puis des requêtes plus précises sur le thème de recherche sont présentées. La recherche des livres pertinents se limite alors à l'exploration de l'ensemble des documents présélectionnés lors de la première étape.
Supposons par exemple le thème de recherche "Gestion". Grâce à la lecture du fichier inversé, le système pourrait répondre que 50 documents sont retenus. Puis la requête "(Gestion) et mini-ordinateurs et coût" pourrait par exemple être présentée, ce qui pourrait entraîner la sélection de deux livres pertinents parmi les 50 qui étaient présélectionnés.

Dans le système Satin 1, un ensemble de pointeurs, constituant le fichier secondaire dynamique, permet de repérer les documents pertinents pour une requête. Ces résultats sauvegardés (à la demande) peuvent être réutilisés pour une requête plus précise, ce qui permet de faire progresser la recherche par étapes successives.
Les recherches se limitent ainsi à un nombre limité de livres. Mais en pratique, on ne peut pas savoir a priori le nombre optimal de livres qu'il faudrait consulter pour satisfaire une requête .

L'introduction d'une syntaxe dans le L.D. ont rendu les traitements informatiques plus complexes mais elle marque cependant une amélioration de la représentation des requêtes et des documents. Prenons par exemple les deux concepts non ordonnés "Plan" et "description". Sans relation indiquant le rôle d'un concept par rapport à l'autre, deux interprétations sont possibles : soit "le plan de la description", soit "la description du plan".
Différents types de relations sont prises en compte, en particulier dans Syntol [20].

Poursuivons encore notre analyse vers les modes de représentation des requêtes qui paraissent meilleurs ou plus évolués : il semble évident que le système ne doit pas seulement atteindre les documents indexés par les mêmes termes que ceux de la requête mais aussi ceux qui paraissent avoir un certain rapport plus ou moins étroits avec cette requête.
Pour atteindre ces documents plus ou moins pertinents, il est indispensable d'amplifier la portée de la requête, en associant des mots clés voisins à ceux qui traduisent la requête initiale, de telle façon qu'une "jonction" puisse se produire entre la représentation de la requête et celle des documents à atteindre. Prenons un exemple pour illustrer cette nécessité :
Soit la requête : "Culture,blé,France".
Soit un document indexé par "Culture,Céréales,Europe".
Pour atteindre ce document, la requête doit être "amplifiée" pour devenir

Culture	blé	France
	ou	ou
	céréales	Europe

Pour effectuer des associations de mots-clés, il faut auparavant connaître les différentes relations qui, dans les systèmes, se résument généralement à l'indication de la synonymie et de la hiérarchie qui existent entre les termes constituant le vocabulaire du L.D.
L'ensemble de ces relations constituent un thésaurus(1).
Dans Satin ou Mistral III/IV, l'indication de l'amplification maximale qu'on autorise dans sa requête suppose auparavant une consultation du thésaurus.
Dans Mistral, cette consultation peut être réalisée grâce à la visualisation du thésaurus déclenchée par l'appel de la procédure d'aide à la recherche M:thésaurus.

---

(1) La définition d'un thésaurus varie suivant les systèmes :
- Dans Satin [16], Syntol [20]et dans [25], un thésaurus est un dictionnaire qui définit les correspondances entre les mots-clés du L.D. et le vocabulaire du L.N.
- Dans le système Mistral [18][19], Smart [27][28], il représente un dictionnaire qui définit des liens sémantiques entre les descripteurs du L.D.
  C'est cette dernière définition que nous donnerons au terme "thésaurus". Les équivalences entre descripteurs du L.D. et les termes du L.N. seront décrites dans le lexique.

Dans Satin, les limites d'amplification sont exprimées à l'aide des opérateurs :

- cheminement(n) qui associe à tout descripteur sa descendance dans l'arborescence du thésaurus
- généralisation(n) qui associe à tout descripteur son ascendance dans l'arborescence (n est la profondeur maximale de l'exploration)
- coïncidence si le lecteur ne veut pas d'approximation.

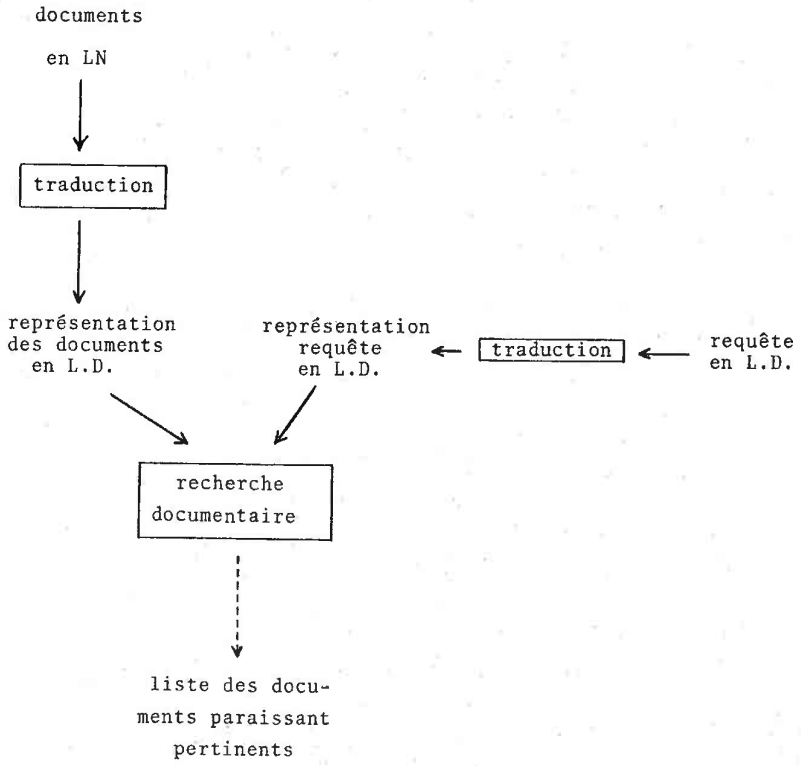
Mais, l'expression d'une requête dans un L.D. entraîne un problème important pouvant intervenir sous deux formes différentes :

- soit le lecteur doit s'adapter au système en apprenant le L.D. qui peut être parfois complexe (au niveau de sa syntaxe, par exemple). Il doit également choisir dans le lexique les mots-clés qui conviennent le mieux pour traduire ses besoins.
- soit une "hôtesse" est spécialement chargée de traduire les désirs des utilisateurs dans le L.D.

Pour éviter l'adaptation des utilisateurs au système ou la présence d'une hôtesse, des recherches ont été effectuées en vue de créer un module de traduction LN → LD.

2.3.2. L'utilisation d'un module de traduction Langage Naturel → Langage Documentaire

En utilisant un tel module, les différents traitements documentaires sont alors les suivants :



La traduction en LD des documents et celle des requêtes ne posent pas les mêmes problèmes :

Les documents sont souvent fort longs. Indexer un document de plusieurs centaines de pages par une dizaine de mots-clés représentatifs est un travail qui est généralement effectué manuellement. L'intervention de l'ordinateur pour traduire des textes courts est encore au stade des essais en laboratoire [10e].

Alors que la traduction d'un document entraîne le problème du "bruit" causé par la surabondance des informations initiales, la traduction d'une requête se heurte au contraire au problème du "silence" provoqué notamment par les informations qui sont sous-entendues dans la requête exprimée. Il ne s'agit plus cette fois de condenser mais plutôt d'amplifier la requête.

Les principaux problèmes rencontrés lors de la traduction automatique des requêtes et des documents en L.D. sont les suivants:

- Il s'agit tout d'abord de reconnaître les différentes formes graphiques que peut prendre un terme donné [10c]. Ces variations sont imposées par les changements de temps, modes, personnes, genres, nombres...

  Exemple : le radical "group" pourrait représenter la famille de mots : Group+age, Group+e, Group+ement, Group+er, Re+group+ement ...

  Mais, rappelons qu'il existe hélas beaucoup de cas tels que celui-ci:

  Art pourrait représenter {Art,Artère,Artiste...}

  Or, "Artère" n'appartient pas à la même famille de mots que "Artiste" [10c].

- Un module de traduction doit également reconnaître les groupes de mots indissociables, c'est-à-dire ceux dont la signification ne découle pas des significations de leurs composants.

  Exemples : chemin de fer, lycée d'Etat, maison d'arrêt ...

- Il doit également être en mesure d'interpréter les termes polysémiques [15]. Ce problème se pose quand la traduction d'un terme du L.N. dans le L.D. varie en fonction du contexte du terme.
  Exemple : groupe est un terme employé en médecine (Groupe sanguin), en mathématiques (groupe abélien), en sociologie (dynamique des groupes)..
  Dans notre exemple, trois classes paramètres sont donc associées à "Groupe". Dans l'expression "groupe le plus souvent commutatif" un balayage à partir du terme "groupe" permet de donner un sens mathématique à ce dernier terme ambigu, lors de la lecture du mot "commutatif".

Une autre approche, de type "statistiques des constructions des phrases", est proposée par Andreewsky. L'exemple suivant est extrait de [10d]:

"Le pouvoir me serait agréable" est une phrase ambiguë. Mais, comme la plupart du temps, en français "le" en début de phrase est article, alors l'algorithme statistique donnera une priorité à l'analyse le = article, pouvoir = substantif. Toutefois, la seconde analyse le = pronom, pouvoir = infinitif ne sera pas rejetée, elle viendra en seconde position.

Toutefois, la traduction L.N. → L.D. pose encore beaucoup de difficultés et la présence d'une hôtesse semble encore indispensable. Le problème posé par les bruits et les silences dans les requêtes persiste notamment.

En fait, il semble qu'un L.D. formel est assez mal adapté pour traduire le flou ou les nuances des requêtes des utilisateurs.

Le langage formel introduit de la rigueur dans la représentation des connaissances et dans l'enchaînement des traitements qui s'ensuivent. Nous voudrions au contraire garder en permanence la diversité d'interprétation du langage naturel et ainsi nous autoriser à effectuer des retours en arrière pour rechercher dans une requête, le niveau de compréhension qui semble être le meilleur, et choisir pour un mot, le rôle paraissant être le plus plausible.

En conservant la diversité d'interprétation du LN, nous supprimons donc la dualité Langage Naturel — Langage Documentaire, ce qui rend notre étude non superposable à celles qui ont permis de réaliser des systèmes documentaires tels que Satin, Smart, Mistral.

3. PRÉSENTATION DE NOTRE ÉTUDE

Nous allons à présent préciser nos objectifs en les situant par rapport aux autres recherches déjà effectuées en documentation automatique.

3.1. Ce que nous avons fait

Nous avons essayé d'atteindre nos objectifs (cf. questions à résoudre en introduction) en utilisant des outils réalisés en Intelligence Artificielle, plus spécialement dans le domaine de la "compréhension" d'énoncés écrits puisque nous désirons interpréter (en totalité ou partiellement) les requêtes des utilisateurs pour rechercher les documents les plus intéressants.

La démarche généralement suivie dans ce cas consiste à confronter le texte à des connaissances qui sont représentées suivant un modèle pré-défini.

Le niveau de compréhension dépend beaucoup de l'application envisagée:

Voici, à titre d'exemple, quelques types de représentation :

- Initialement, les textes étaient le plus souvent traduits à l'aide de graphes de nature syntaxique [3]. Le but recherché était alors de déterminer la structure profonde d'une phrase. Nous verrons au chapitre III qu'un analyseur syntaxique peut avoir une utilité plutôt limitée en documentation automatique.

- Schank,dans le projet Margie [9] considère qu'un texte peut être représenté par un graphe ("grammaire de dépendance conceptuelle") dans lequel chaque action est traduite par une combinaison de primitives.

-Pour COULON et KAYSER [6], comprendre un texte, c'est trouver pourquoi il est plausible, dans le contexte où il se trouve. Prenons un exemple :
"Il fait jour à minuit" est une proposition peu plausible. Nous dirons ne pas la comprendre, sauf si, par ailleurs, il est écrit dans l'énoncé que la scène se passe au-delà du cercle polaire et qu'il y a parmi nos connaissances se rapportant à cet endroit la possibilité de voir le soleil à minuit.

Nous pensons que ce type de raisonnement pourrait s'appliquer en documentation automatique, moyennant certaines modifications. Supposons la requête :
"Auriez-vous des livres sur la gestion des bases de données?"
Transformons cette question en une affirmation :
"Vous avez des livres sur la gestion des bases de données".
Il reste à évaluer la plausibilité de cette proposition en testant sa proximité par rapport à la représentation des livres du fonds.

Parmi les traitements sur la compréhension d'énoncés en général, nous allons à présent préciser ceux qu'il faudrait a priori privilégier, compte tenu de la spécificité de la documentation automatique.

Définir diverses manières d'établir des inférences à partir d'une requête sera plus particulièrement une de nos motivations majeures : contrairement à un texte, une requête est en effet fréquemment courte et imprécise. Il faut donc souvent essayer de l'enrichir, afin d'amplifier sa portée, ce qui peut permettre d'atteindre des livres plus ou moins pertinents.
Pour effectuer cette amplification, nous essayerons de ne plus nous limiter à l'utilisation des relations de synonymie et de hiérarchie entre des descripteurs.

L'idée des "frames" (MINSKY [10f]) nous paraît intéressante pour simuler le comportement d'un documentaliste et guider les recherches documentaires : pour appréhender les besoins d'un lecteur, le documentaliste mène une conversation qui évolue de façon cohérente, au-delà de son aspect quelquefois décousu. En ce qui concerne le lecteur, il peut exprimer dans sa requête le désir de faire ultérieurement une activité. C'est grâce à la connaissance du déroulement probable de cette activité, que l'on peut prévoir ses besoins et établir les objectifs de la recherche documentaire.

Pour amplifier la portée d'une requête, nous utiliserons en particulier un réseau sémantique dans lequel la proximité de 2 termes du Langage Naturel sera valuée.
D'une façon plus précise, ce réseau permettra :

- de préciser le sens d'un terme d'après son contexte
- de rendre capable le système de suggérer et de contrôler lui-même l'amplification de la portée des requêtes, ce qui éviterait la consultation d'un thésaurus pour définir les limites d'amplification.
- d'évaluer progressivement la pertinence des documents présentés :

. par rapport à chacun des objectifs de la recherche définis dans la requête puis
. par rapport à l'ensemble de la requête.

Les résultats édités vont ainsi faire apparaître les différents objectifs plus ou moins atteints par la recherche documentaire. Dans la plupart des autres systèmes documentaires (exemple : Mistral), les utilisateurs effectuent généralement un dernier filtrage des documents proposés, en lisant leur résumé. Avec notre approche, le lecteur pourrait être en mesure de filtrer les documents, en découvrant sur l'état édité ceux qui traitent le mieux les objectifs les plus importants de sa requête.

Par analogie et par abus de langage, nous donnerons le nom de "thésaurus" à ce réseau sémantique.

Mis à part le thésaurus, les autres connaissances seront principalement représentées par des procédures parmi lesquelles nous distinguerons deux grands types principaux :

- les procédures stratégiques dont le but principal est la détermination de l'esprit dans lequel une recherche documentaire doit être effectuée. Elles sont utilisées pour l'amplification des requêtes.
- les procédures documentaires destinées à représenter des livres du fonds documentaire (contenu, intérêt,...).

Une procédure -stratégique ou documentaire- est un questionnaire en forme d'arborescence dont les noeuds sont des objectifs et dont les feuilles sont des actions.

Dans le cas d'une procédure stratégique, les actions permettent d'enrichir la requête par de nouveaux objectifs. Dans le cas d'une procédure documentaire, les actions déclenchées sont des propositions de livres pertinents.

Nous décrirons plus précisément la structure et le mode d'exploration de ces deux types de procédures (pages 82 et 95).

Pour expliquer la nature des informations contenues dans ces deux classes de procédures, nous allons prendre un exemple :

Supposons la requête : "j'aimerais savoir comment on programme". Si seul le concept "programmation" est retenu, la représentation de la requête devient imprécise car la programmation est un domaine documentaire très vaste. Plusieurs documents se rapportant à la programmation risquent alors de ne plus vraiment être pertinents.

Les termes "savoir" et "comment" doivent suggérer au système de rechercher des livres dans lesquels sont décrits des exemples de programmes ainsi qu'une méthodologie simple. Il semble également intéressant de donner au lecteur des conseils pour bien programmer.

La connaissance du type d'utilisateur ayant présenté cette requête est également très utile pour guider le système dans sa recherche documentaire : supposons qu'il s'agisse d'un utilisateur non-informaticien. Le système peut déduire de cette information que l'utilisateur est probablement un débutant en informatique qui désire s'initier à la programmation.

Aucun langage n'étant précisé par l'utilisateur, le système pourrait même lui suggérer de programmer en Pascal, par exemple.

Les procédures stratégiques ont pour but de réaliser ce type d'inférences qui permettent, dans notre exemple, de ne pas représenter la requête par "programmation", mais par la liste d'objectifs[1] : programmation, exemples, méthodologie, conseils,

(1) En documentation automatique, un descripteur est un terme dont la signification est unique et qui sert à décrire un document. A la place du terme "descripteur", nous emploierons désormais le terme "objectif" pour les raisons suivantes :
- contrairement à un descripteur, un objectif pourra avoir diverses significations
- un objectif sera un terme du LN qui pourra être utilisé pour décrire des documents, mais aussi des lecteurs, des requêtes...
- "objectif" a été le terme choisi car les termes significatifs d'une requête représentent les objectifs de la recherche documentaire. Les termes qui décrivent les livres peuvent également être considérés comme des objectifs à atteindre s'ils se rapportent suffisamment à la requête et s'ils mènent à des livres pertinents.

non-informaticien, initiation, débutant...

Cette représentation enrichie de la requête rend plus explicites les besoins du lecteur qui pourraient par conséquent, être mieux appréhendés.

Cet exemple montre aussi que dans un document, d'autres informations que le sujet abordé sont également très importantes: certains livres paraissent hermétiques, d'autres au contraire, vulgarisent des travaux complexes. La présence d'exemples, de schémas explicatifs, d'exercices éventuellement avec leur corrigé, d'applications, de conseils, de critiques, la clarté, le style... sont par exemple des informations qui peuvent être voulues par un lecteur.

L'important n'est donc pas vraiment le contenu mais ce qui va nous décider à proposer un livre.

Pour définir les conditions plus ou moins importantes à remplir d'une façon suffisamment satisfaisante pour atteindre automatiquement ce livre, le bibliothécaire[1] devra donc établir des jugements assez précis sur les diverses caractéristiques d'un livre : en fonction de la confiance qu'il accordera aux différentes informations contenues dans le livre, il devra discerner les qualités du livre, de telle façon qu'il puisse le démarquer par rapport aux autres et le suggérer pour un type de besoins bien particuliers. Après l'analyse des documents ainsi faite, il doit ensuite faire une description du domaine analysé. A un domaine va alors correspondre une procédure documentaire définissant les lignes de conduite permettant d'atteindre les documents.

3.2. Ce qui pourrait être fait

Nous avons voulu avant tout définir et tester un modèle de représentation d'une partie des connaissances d'un documentaliste.

Après avoir atteint ces objectifs, nous pourrons alors envisager de rendre conversationnel le système créé, ce qui le rendrait probablement plus performant.

---

(1) Un bibliothécaire a plutôt un rôle de supervision, de gestion et d'administration de sa bibliothèque tandis qu'un documentaliste a plutôt un rôle de conseiller auprès des lecteurs.
Il se peut que nous confondions quelquefois les deux termes vu qu'il arrive qu'une même personne remplisse ces deux fonctions.

Après cette présentation de nos objectifs, nous allons à présent décrire comment sont analysées les requêtes, puis nous montrerons comment est représenté le fonds documentaire et quelle est la stratégie de recherche documentaire employée.

III. L'ANALYSE

AUTOMATIQUE

DES REQUETES

III. L'ANALYSE AUTOMATIQUE DES REQUETES

1. LE FILTRAGE DES OBJECTIFS D'UNE REQUETE

Le système documentaire ne peut travailler qu'à partir des termes d'une requête sur lesquels il a des connaissances. Un bagage culturel linguistique est donc nécessaire au système pour comprendre les requêtes et le fonds documentaire.

Un lexique contient tous les objectifs pouvant servir à l'indexation des requêtes et des documents.

Pour filtrer les objectifs d'une requête, le système lit mot à mot cette requête : les termes qui appartiennent au lexique sont inclus dans la liste des objectifs de la requête. (Pour les différencier des autres termes, ils seront désormais soulignés dans chacune des requêtes présentées).

Si des renseignements sur les lecteurs sont connus par le système, ils sont également considérés comme objectifs. Dans les bibliothèques dont la gestion du prêt est automatisée, des fichiers lecteurs sont déjà créés et pourraient donc être utilisés lors des recherches documentaires.

Certaines connaissances générales sont également très utiles pour sélectionner les documents les plus pertinents. Dans ce cas, elles doivent aussi être intégrées dans la liste des objectifs.

Exemples d'objectifs :

- Un professeur présente la requête : "Je voudrais un livre d'exercices de PL/1"?
- Un étudiant demande : "Comment organiser les données d'un fichier"?

Remarque :

La richesse du vocabulaire dans le lexique augmente les potentialités du référentiel auquel le système documentaire recourt pour interpréter le code graphique et le doter de significations. Mais plus ce référentiel est abondant, plus la place occupée dans la mémoire est grande et plus la durée de l'activité exploratrice dans le lexique est longue.

2. L'IMPORTANCE DES DIFFERENTS OBJECTIFS D'UNE REQUETE

Les objectifs d'une requête sont plus ou moins importants. Cependant, ils ne sont pas pondérés car leur importance est très difficile à déterminer :

Une pondération manuelle est à éviter car il s'agit d'un traitement trop subjectif : la valeur d'un poids à attribuer à un objectif d'une requête donnée varie suivant les lecteurs. De plus, cela nécessiterait une réflexion de la part des utilisateurs, pour analyser leur requête : c'est au système d'effectuer cette analyse.

Il n'est pas non plus toujours exact de considérer que plus la fréquence moyenne d'apparition d'un objectif dans un texte est élevée, plus cet objectif est important ([17]). Une autre école diamétralement opposée considère que ce sont les concepts à fréquence rare qui sont les plus importants.

Nous considérons que tous les objectifs d'une requête sont a priori importants et doivent être atteints au terme de la recherche documentaire. En conséquence, une exploration du fonds documentaire ne s'arrêtera que lorsque l'ensemble des livres proposés seront suffisamment pertinents par rapport à chacun des objectifs de la requête.

2.1. Influence d'un ordre de priorité des objectifs sur le déroulement de la recherche documentaire

Il y a cependant de multiples façons de parvenir à la satisfaction d'une requête. En particulier, l'ordre de priorité établi dans la liste des objectifs d'une requête influence le déroulement de la recherche des documents pertinents ; la recherche s'arrête dès qu'un certain nombre de documents intéressants est trouvé. Afin qu'elle soit plus logique et plus rapide, il est souhaitable de rendre prioritaires les objectifs qui paraissent être les plus importants.
Cet ordre de priorité sera établi grâce à une analyse syntaxique sommaire de la requête.

2.2. Etablissement d'un ordre de priorité entre les objectifs d'une requête grâce à son analyse syntaxique (1) :

2.2.1. Définition de l'analyse syntaxique

Définir le terme "syntaxe" n'est pas simple, sa frontière avec la sémantique étant difficile à préciser [6].

Pour ordonner les objectifs, il nous semble suffisant de repérer leurs occurrences dans la requête, leur ordre d'apparition et leurs relations matérialisées éventuellement par des prépositions.

2.2.2. But de l'analyse syntaxique

Dans un premier temps, la requête est décomposée en son thème, d'une part, et le contexte du thème, d'autre part.

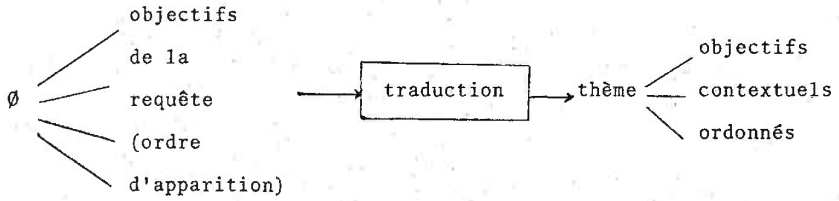
Le thème est l'objectif prioritaire de la recherche : entre tous les objectifs représentant une requête, c'est celui qui marque l'objet le plus central de la recherche, ou son "foyer" le plus important.

Le contexte du thème est formé des objectifs de la requête, moins le thème.

---

(1) L'analyse syntaxique proposée se veut très sommaire. Sa fonction essentielle est de dégrossir la requête, et des erreurs d'interprétation peuvent s'y glisser. Il restera à la charge des autres modules de compléter ou de corriger l'imbrication des objectifs de la requête.

La requête est traduite en un graphe ayant pour racine le thème :



Ce graphe permettra d'établir l'ordre de priorité recherché.

"Ø" signifie que le thème est initialement indéterminé.

2.2.3. Principe d'analyse syntaxique

L'analyse syntaxique permet d'élaborer des hypothèses -les plus probables d'abord- quant à la structure et au sens à donner à une partie de requête étudiée.

Puis ces hypothèses ayant été élaborées, il s'agit de les vérifier en les confrontant à un premier sens général déjà envisagé et qui doit servir de guide aux investigations de l'analyse :

Le système doit donc formuler une hypothèse, établir une stratégie de vérification et conclure à une structure de la partie analysée. L'analyse se fait donc par approximations successives, par élaborations d'hypothèses de mieux en mieux confirmées.

2.2.4. Règles d'analyse syntaxique

D'après l'appartenance des objectifs à certaines catégories syntaxiques ou à certaines classes sémantiques(1), le système va essayer de déterminer la structure d'une requête.

Il s'agit par conséquent d'une analyse par cas. En voici quelques exemples :

préposition en :

Si on a obj_i en obj_j et

(1) Si obj_j est une date ⇒ obj_i —— obj_j/date

Ceci signifie qu'après traduction, obj_i est le thème par rapport à obj_j. L'indication portée après le "/" exprime ce que représente obj_j par rapport à obj_i.

exemple : les mini-ordinateurs en 1980

(2) Si obj_j est un lieu ⇒ obj_i —— obj_j/lieu

exemple : l'informatique en Italie

(3) Si obj_i est un traitement et si obj_j est un mode d'exploitation

⇒ obj_i —— obj_j/manière d'être

exemple : contrôle en temps réel

(4) Si obj_i est une activité et si obj_j est une langue ou un langage

⇒ obj_i —— obj_j/manière

exemples : traduction en anglais

programmation en Pascal

(1) La représentation des liens sémantiques entre des objectifs du lexique (exemple : appartenance d'un objectif à une classe) sera décrite ultérieurement (cf. chapitre "le thésaurus", p. 55).

(5) Si obj_i est un déplacement et si obj_j est un moyen de transport

⇒ obj_i —— obj_j/moyen

exemple : un voyage en avion

(6) Si obj_i est un objet et si obj_j est une matière

⇒ obj_i —— obj_j/constitution

exemple : un buste en marbre

(7) Si obj_i est un travail et si obj_j est une durée

⇒ obj_i —— obj_j/durée

exemple : compilation en 30 secondes

"En" en corrélation avec "de" (obj_i de obj_j en obj_k)

(8) Si obj_i est une action et si obj_j et obj_k sont des objets

⇒ obj_i — obj_j/progression, obj_k/progression

exemple : aller de fleur en fleur

(9) Si obj_j est une durée et si obj_k est une durée

⇒ obj_i — obj_j/périodicité, obj_k/périodicité

exemple : empirer d'heure en heure

...

Remarques :

-Une partie condition qui figure après un "/" n'est pas décrite par un seul objectif mais aussi par ses semblables que l'on peut atteindre à travers un réseau sémantique (cf. chapitre : "le thésaurus" p.55).

- Nous verrons que le travail de la thèse ne porte pas vraiment sur l'analyse syntaxique des requêtes mais surtout sur la façon de conduire certaines formes d'inférences afin de pouvoir répondre à ces requêtes.
- Nous continuons de donner quelques règles dans d'autres cas. Il conviendrait de les affiner.

Préposition sur : obj_i sur obj_j

(10) Si obj_i est un traitement informatique et si obj_j est un outil informatique ⇒ obj_i —— obj_j/moyen

exemple : la comptabilité sur un mini-ordinateur

...

Préposition de : obj_i de obj_j

(11) Si obj_i est une activité ou une action ⇒ obj_j —— obj_i

exemple : l'analyse d'un texte

(12) Si obj_i est un outil et si obj_j est une activité

⇒ obj_j —— obj_i/moyen

exemple : un mini-ordinateur de gestion

...

Préposition par : obj_i par obj_j

(13) Si obj_i est un traitement et si obj_j est un outil

⇒ obj_i —— obj_j/moyen

exemple : traduction par le compilateur

...

Préposition dans : obj_i dans obj_j

(14) Si obj_j est un lieu ou un objet ⇒ obj_i —— obj_j/lieu

exemple : l'informatique dans les petites entreprises

Très souvent, des prépositions ne figurent pas entre les objectifs. Nous utilisons alors des règles de ce type :

Soient N = nom

V = verbe

adv = adverbe

(15) Si V_1V_2N ⇒ $\underbrace{N — V_2 — V_1}_{V}$

exemple : savoir réaliser un programme

⇒ programme —réaliser —savoir

Le "V" situé en dessous de l'accolade signifie que le groupe $N—V_2—V_1$ pourrait jouer le même rôle qu'un verbe dans le cas où un autre objectif est à relier à ce groupe.

(16) Si adv V N ⇒ N—V—adv
V

exemple : comment rédiger un programme
⇒ programme — rédiger — comment

(17) Si V adv ⇒ V—ady
V

exemple : savoir où ⇒ savoir——où

2.2.5. Exemples d'analyse syntaxique

requête n°1 :

Description [des] mini-ordinateurs [de] gestion

Un thème est déterminé parmi les deux premiers objectifs. D'après la règle n° (11) (page 45), "mini-ordinateur" est le thème :

mini-ordinateur ——description

L'objectif suivant (gestion) est ensuite pris en compte. Le système cherche comment il peut intégrer ce nouvel objectif dans la partie de la requête déjà analysée. D'après la règle n° (12), on obtient la structure :

gestion—— mini-ordinateur —— description

requête n°2 :

Description [de] l'automatisation [du] prêt [dans] les bibliothèques

La règle n° (11) permet de considérer "automatisation" comme étant le thème virtuel.

Pour chaque nouvel objectif, on analyse son rôle par rapport au thème virtuel des objectifs déjà pris en compte.

D'après la règle n° (11), on peut écrire :

prêt —— automatisation —— description

Le rôle du nouvel objectif "bibliothèque" permet de confirmer que "prêt " est le thème :

prêt — automatisation —— description
prêt — bibliothèque/lieu

requête n°3 :

Calcul [des] pondérations [d'] attribution [de] crédits [dans] une banque [en] 1980

D'après la règle n° (11), pondération est le thème des deux premiers objectifs

pondération ___ calcul

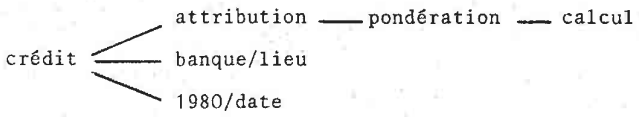
Après la prise en compte de l'objectif suivant (attribution), la structure obtenue grâce à la règle n° (11) est la suivante :

attribution ___ pondération ___ calcul

Puis, toujours grâce à la règle n° (11), "crédit" devient le thème :

crédit ___ attribution ___ pondération ___ calcul

Le rôle, par rapport au thème virtuel actuel, du nouvel objectif "banque", puis celui de l'objectif "1980", déterminés respectivement grâce aux règles n° (14) et n° (1) permettent de confirmer que crédit est le thème définitif de la requête :



2.2.6. Règles de priorité des objectifs d'une requête

Le thème est l'objectif le plus prioritaire. D'une façon générale, plus un objectif est situé profondément dans le graphe obtenu au terme de l'analyse syntaxique, moins cet objectif est prioritaire.

Pour les objectifs situés à un même niveau de profondeur dans un graphe, ce sont les objectifs qui indexent le plus de livres qui sont considérés les plus prioritaires.

La raison est la suivante :

Soit n la taille de l'ensemble des livres indexés par un objectif. n reflète la richesse de cet ensemble, ou, si les livres de la bibliothèque sont bien adaptés aux besoins de ses utilisateurs, la fréquence d'apparition de cet objectif dans les requêtes donc approximativement l'importance que les lecteurs lui accordent généralement.

Exemple : pour la requête n°3, si les nombres des livres indexés par "attribution", "banque", "1980" sont respectivement 2, 12, 8, l'ordre de priorité des objectifs de la requête sera : crédit, banque, 1980, attribution, pondération, calcul.

2.2.7. Nécessité et limites d'un analyseur syntaxique

Un analyseur syntaxique est nécessaire car, dans certains cas, c'est la syntaxe d'une phrase qui permet de la comprendre.
Exemples :

Supposons que "analyse", "méthode", "exploitation", "système" soient quatre objectifs appartenant au lexique.

(1) Les deux termes non ordonnés "méthode" et "analyse" peuvent signifier "analyse d'une méthode" ou "une méthode d'analyse" (informatique de gestion)

(2) De la même façon, si "exploitation" et "système" ne sont pas ordonnés, ils peuvent signifier "exploitation d'un système" ou "un système d'exploitation".

Dans ces deux exemples, c'est l'ordre des mots qui permet de choisir la signification réelle.

Un analyseur syntaxique a cependant une utilité limitée pour plusieurs raisons :

- Il est difficile d'en construire un pour l'ensemble des requêtes en L.N. car elles peuvent s'exprimer de multiples façons.
- Il arrive assez souvent que les données sémantiques entre deux termes particuliers ordonnés imposent un rapport unique que l'on découvre à travers le réseau sémantique (cf. chapitre "le thésaurus" p. 55).

Exemples : mini-ordinateur ___ coût
programmation ___ cobol
anglais ___ syntaxe

- Par ailleurs, certains objectifs ou certaines combinaisons d'objectifs ne sont plus ambigus quand on connaît leur contexte, c'est-à-dire l'ensemble des objectifs qui "entourent" le ou les objectifs dont l'interprétation n'est pas certaine.

Exemples :

."analyse fonctionnelle et organique" se rapporte probablement à l'informatique de gestion

."exploitation, système, recherche,documentaire"devra plutôt être interprété "exploitation d'un système documentaire pour la recherche d'informations", surtout si l'utilisateur est un chercheur en informatique documentaire.

- De plus, la connaissance des objectifs qui indexent une requête, de la structure de cette requête et de la nature des rapports existant entre les objectifs ne signifie pas que la requête est comprise par le système documentaire :
La compréhension d'une requête par le système est testée par la qualité de ses réponses. Or les informations contenues dans les documents ne répondent le plus souvent qu'indirectement à une question

Exemple :

Requête : "Auriez-vous des livres se rapportant à l'étude quantitative de diverses stratégies de gestion de l'unité centrale, de la mémoire principale et de la mémoire secondaire d'un système en temps-partagé"?
Le système doit pouvoir sélectionner les deux documents suivants, bien que ceux-ci ne soient pas exactement décrits par les mêmes objectifs que la requête :

. Optimisation des transferts entre mémoire centrale et disques.
. Simulation de l'allocation de ressources dans un système conversationnel à mémoire virtuelle paginée.

Il convient donc d'établir un outil permettant une analyse plutôt sémantique des requêtes.

Grâce à cet outil, le système devra comprendre un objectif d'après son contexte et surtout être capable de déterminer la totalité des objectifs qui expriment plus ou moins une requête.

3. AMPLIFICATION DE LA PORTÉE D'UNE REQUETE

Le système documentaire doit satisfaire des requêtes auxquelles les documents ne répondent pas directement.
L'amplification de la portée d'une requête consiste à associer aux objectifs qui traduisent cette requête, des objectifs qui leur sont plus ou moins proches sémantiquement : il ne faut pas oublier que toute information ne peut pas être connue par l'ordinateur si elle ne lui est pas transmise explicitement.
Pour détecter les documents pertinents, le système devra repérer les objectifs qui sont symbolisés par les mêmes signes graphiques dans la représentation de la requête présentée et dans celle des documents.
L'amplification de la portée d'une requête sera réalisée grâce au réseau des approximations appelé "thésaurus" et grâce aux connaissances stratégiques, organisées en "procédures stratégiques" destinées à connaître l'esprit dans lequel les recherches documentaires doivent être effectuées.

3.1. Particularités de l'amplification des requêtes présentées à notre système

En mode conversationnel, le réseau des approximations peut être visualisé, ce qui permet au lecteur de choisir tous les objectifs qui traduisent le mieux ses besoins [18][19]. Dans le système Satin [16], c'est le lecteur qui indique au système les approximations maximales qu'il est autorisé à effectuer éventuellement.

Notre système calcule la distance sémantique existant entre deux objectifs quelconques, ce qui le rend capable de suggérer et de contrôler lui-même l'amplification des requêtes.

Par ailleurs, nous avons déjà vu, lors de la présentation de notre étude, qu'il est utile pour le système documentaire d'essayer de connaître l'esprit dans lequel une recherche doit être effectuée. Par exemple, le type de livre à conseiller à un chef de projet n'est généralement pas le même que celui qui doit être sélectionné pour un pupitreur, leurs besoins étant le plus souvent différents.

Les informations générales telles que "se renseigner", "vouloir connaître", "comment", "pourquoi" sont également précieuses dans la mesure où elles permettent de comprendre ce que veut le lecteur à propos du sujet exprimé dans sa requête.

3.2. L'amplification de la portée d'une requête grâce à un thésaurus

3.2.1. Le thésaurus

3.2.1.1. Rôle

Le thésaurus sert à exprimer des liens sémantiques existant entre les différents objectifs du lexique.

Les requêtes et les livres étant indexés par ces objectifs, l'évaluation de la pertinence des livres proposés pourra s'effectuer grâce au thésaurus. C'est aussi grâce à lui que le système pourra satisfaire des requêtes auxquelles les documents ne répondent pas directement. Le thésaurus permettra par conséquent d'établir des recherches exhaustives sur un sujet donné.

3.2.1.2. Description des liens sémantiques

forme d'un lien :

Une affinité sémantique entre deux objectifs sera exprimée de la façon suivante :

(objectif a, objectif b) relation R_i, plausibilité pl/objectif

Ceci signifie qu'il existe une relation R_i entre l'objectif a et l'objectif b. Cette relation peut-être plus ou moins vraie : une valeur appelée plausibilité d'intérêt donne une mesure à la relation entre les deux objectifs.

Si l'objectif c est présent, il désigne le domaine de validité en dehors duquel la relation entre l'objectif a et l'objectif b n'est plus vraie.

Pour être plus précis, nous allons étudier successivement, les relations, les plausibilités et les domaines de validité.

3.2.1.2.1. Les relations

Les types de relation autorisés sont les suivants :

① Voisin (abréviation vd)

(a,b) vd ⇔ a et b désignent un concept similaire

a → b, b → a (il est possible d'associer a à b ou b à a)

exemple : (bateau, navire) vd

(a,b) vdi (voisinage dissymétrique)

a → b (il est seulement possible d'associer b à a)

exemple : (rédaction,réalisation) vdi dans le cas de la rédaction d'un dossier

② Sorte-de (abréviation sd)

(a,b) sd ⇔ (b,a) sd^{-1} ⇔ a est un objet du type défini par b

a → b, b → a

exemples : (chat,mammifère) sd

(smart,système-documentaire) sd

(temps-partagé,mode d'exploitation) sd

③ Partie de (abréviation pd)

(a,b) pd ⇔ (b,a) pd^{-1} ⇔ la notion exprimée par a est incluse dans celle définie par b

a → b, b → a

exemples : (volant,voiture) pd

(compilateur,logiciel) pd

④ Moyen

(a,b) moyen ⇔ (b,a) $moyen^{-1}$ ⇔

a est un moyen de réaliser b

a → b, b → a

exemples : (corig,automatisation) moyen

⑤ associé à (abréviation : ass)

(a,b) ass ⇔ association de a à b (ou de b à a)

(a → b, b → a)

Cette relation est employée quand aucune des autres relations ne convient à l'expression d'un lien. Elle désigne un rapport beaucoup plus vague.

(a,b) asdi (association dissymétrique)

a → b

exemples : (personnalité,culture) ass

(médaille,mérite) ass

(deuil,noir) asdi

(mistral,violent) asdi

Les différents types de relation sont rangés dans le tableau RELATION

Remarque : Nous aurions pu prendre en compte d'autres types de relations telles que celles qui figurent dans syntol [20] mais nous considérons que ce n'est pas très profitable dans un premier temps.

3.2.1.2.2. Les plausibilités de proximité (cf.COULON,KAYSER [8])

Il existe plusieurs algorithmes de détection automatique de voisinage sémantique. Nous avons utilisé celui qui est proposé par COULON-KAYSER [8] pour définir nos algorithmes d'évaluation de proximités sémantiques, en essayant de les adapter à la documentation automatique.

La proximité entre deux objectifs a et b quelconques est mesurée grâce à une plausibilité de proximité notée plprox(a,b). D'une façon générale, une plausibilité est un nombre réel appartenant à l'intervalle [0,1].

- Par convention, plus l'éloignement sémantique entre deux objectifs a et b est grand, plus la plausibilité de proximité plprox(a,b) tend vers 0.
  Exemple : plprox(chaud,froid) = 0.05
- Au contraire, plus l'éloignement sémantique entre deux objectifs a et b est faible, plus la plausibilité de proximité plprox(a,b) tend vers 1.
  Exemple : plprox(navire,bateau) = 0.95
- Au milieu de ces deux bornes 0 et 1, la valeur 0.5 signifie qu'il y a incertitude totale sur le degré d'affinité des 2 objectifs.
  Exemple : si le système ne sait pas qu'un casoar est un oiseau, plprox(casoar,oiseau) = 0.5

Nous allons voir dans ce chapitre que la plausibilité mesurant la proximité (plprox) entre deux objectifs est calculée à l'aide de deux autres types de plausibilité :

- la plausibilité relationnelle pl_r liée à la nature de la relation R_i entre les deux éléments
- la plausibilité d'intérêt pl qui mesure l'opportunité de cette relation pour les deux éléments.

3.2.1.2.2.1. Composition des plausibilités de proximité [8]

Sur cet ensemble PL des plausibilités de proximité, deux lois de composition internes notées S (Série) et // (Parallèle) ont été définies. Elles ont pour but respectif d'affaiblir et de renforcer les plausibilités de proximité.

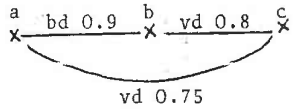
Parmi les axiomes qui définissent ces lois, on trouve en particulier :

1 S x = x ; 0.5 S x = 0.5 ; 1 // x = 1 ; 0.5 // x = x

Exemples :

- Série : Une plausibilité de proximité est affaiblie si un éloignement sémantique croît.

  exemples : (a,b) vd 0.9 } (a,c) vd 0.9 S 0.8 = 0.75
  (b,c) vd 0.8 }



L'éloignement sémantique entre a et c est égal à celui existant entre a et b augmenté de celui qui figure entre b et c

- Parallèle : Une plausibilité de proximité est renforcée si un éloignement sémantique décroît.

exemple : Supposons qu'on ait à déterminer la plausibilité de proximité entre le livre L1 et l'objectif d'une requête "multiprogrammation". Initialement, il y a incertitude totale sur le rapport :

plprox(L1,multiprogrammation) = 0.5

Supposons qu'on découvre que L1 se rapporte à la description de modes d'exploitation et que

plprox(multiprogrammation,mode d'exploitation)=0.7

Le rapport entre L1 et multiprogrammation est accru :

plprox(L1,multiprogrammation) = 0.5//0.7 = 0.7

Puis, supposons qu'on s'aperçoive que L1 décrit l'allocation de temps et que plprox(multiprogrammation, allocation de temps)= 0.8. Cette deuxième information tend à confirmer l'hypothèse d'un rapport entre L1 et multiprogrammation

plprox(L1,multiprogrammation) = 0.7 // 0.8 = 0.9

Notre conviction que L1 est indexé par multiprogrammation est accrue. La plausibilité de proximité est donc renforcée.

3.2.1.2.2.2. Les plausibilités relationnelles

Dans un lien sémantique (a,b) R_i pl, l'éloignement sémantique entre a et b varie suivant le type de la relation R_i et suivant la plausibilité d'intérêt pl. Pour indiquer que les relations désignent des rapports plus ou moins étroits entre les objectifs qu'elles relient, une plausibilité relationnelle, notée pl_r , est attribuée à chacune de ces relations.

Le tableau PLREL contient ces plausibilités relationnelles.

relations		plausibilités relationnelles
noms	abréviations	
voisin de	vd	1
voisin dissymétrique	vdi	1
sorte de	sd	0.8
(sorte de)$^{-1}$	sd^{-1}	0.9
partie de	pd	0.8
(partie de)$^{-1}$	pd^{-1}	0.9
moyen	moyen	0.9
moyen^{-1}	moyen^{-1}	0.9
associé à	ass	0.8
associé dissymétrique	asdi	0.8

Tableau des relations et de leur plausibilité relationnelle

Pour chaque relation R_i appartenant au tableau RELATION, la plausibilité relationnelle a été fixée a priori d'après l'éloignement sémantique estimé entre deux objectifs reliés par R_i. On peut voir dans le tableau PLREL que nous avons donné par exemple une légère préférence à la généralisation (sd^{-1}) plutôt qu'à la particularisation (sd) : faute de trouver un livre répondant strictement à l'objectif recherché on peut compter sur le lecteur pour rechercher et découvrir l'information dont il a besoin dans un livre traitant d'un problème plus global. D'autre part, il n'est pas malsain de conseiller à quelqu'un d'approfondir certains aspects plus étroits, mais dans ce cas, on prend le risque de ne pas avoir bien focalisé son centre d'intérêt.

3.2.1.2.2.3. Les plausibilités d'intérêt

Elle sert à indiquer que la relation entre deux objectifs est plus ou moins valide, suivant les objectifs envisagés.

Exemple : (clair, foncé) vd 0.05

On ne mémorise pas des connaissances a priori fausses car on n'a pas l'occasion de les utiliser.D'une façon générale une plausibilité d'intérêt entre deux objectifs a et b est faible (pl ≈ 0.5) si on considère qu'il n'est pas intéressant d'associer l'objectif a à l'objectif b.

Exemple : (Pascal,Fortran) vd 0.55

Car si quelqu'un demande un livre de Pascal on ne va pas

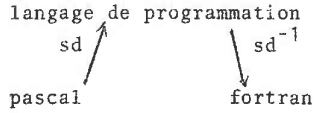
lui donner du Fortran même si c'est aussi un langage de programmation[1].

3.2.1.2.2.4. Calcul de la plausibilité de proximité de deux objectifs reliés directement

La plausibilité relationnelle pl_r et la plausibilité d'intérêt pl contribuent à affaiblir la plausibilité résultante plprox.

plprox(a,b) = plausibilité relationnelle de R_i S plausibilité d'intérêt pl

(1) Nous nous heurtons ici au problème de la transitivité :



Pour éviter d'associer indirectement un langage de programmation à un autre, les plausibilités d'intérêt des liens sémantiques concernés doivent aussi être suffisamment faibles.

3.2.1.2.3. Domaine de validité

Quand il est plus petit que l'ensemble des connaissances du système, il est à placer à la fin des liens sémantiques.

Exemple : (a,b) R_i pl / c ← domaine de validité

C'est un objectif qui désigne le plus souvent une matière en dehors de laquelle la plausibilité de proximité entre les deux objectifs a et b n'est plus toujours vraie. Il définit donc les conditions d'association de b à a (ou éventuellement inversement de a à b).

N'importe quel objectif peut être utilisé comme domaine de validité.

Le concept "domaine de validité" permettra de résoudre partiellement les problèmes de polysémie.

Exemples :

-(syntaxe,analyse) moyen 0.9 / intelligence artificielle

"syntaxe" ne sera associé à "analyse" que si la requête a un rapport suffisant avec l'intelligence artificielle

- (analyse,conception) moyen 0.9 / gestion

"analyse" ne sera associé à "conception" que si la requête se rapporte suffisamment à l'informatique de gestion.

Remarque : Les domaines de validité pourraient dans une certaine mesure, donner la possibilité au système documentaire de simuler l'activité d'un documentaliste appartenant à une bibliothèque d'une grande taille, avec une matière documentaire et un ensemble de lecteurs hétérogènes.

Exemples : (coordination,activité) sd 0.85 / chef de projet

(manipulation,activité) sd 0.85 / opérateur

"coordination" et "manipulation" ne peuvent pas être substitués à "activité", quelque soit le type de lecteur.

3.2.1.3. Exemples de liens sémantiques dans un thésaurus

① (comptabilité,application) sd 0.6 / gestion
② (conception,automatisation) moyen 0.9 / gestion
③ (réalisation,automatisation) moyen 0.9 / gestion
④ (analyse, concevoir) moyen 0.9 / gestion
⑤ (programmation,réalisation) moyen 0.9 / gestion

Justification :

Dans le domaine de l'informatique appliquée à la gestion (gestion)

① Comptabilité est une application particulière
Pour automatiser une application, il faut
② concevoir puis
③ réaliser les traitements informatiques correspondant à cette application
④ les traitements sont conçus au terme d'une analyse (fonctionnelle puis organique)
⑤ la programmation consiste à réaliser les programmes informatiques qui simuleront l'activité correspondant à l'application (exemple : comptabilité)

3.2.1.4. Mode de construction du thésaurus

Le thésaurus pourrait être construit manuellement ou semi-automatiquement[1].

---

(1) La construction semi-automatique consiste à effectuer, dans un premier temps, une analyse automatique sur les co-occurrences des termes dans des textes d'un corpus, puis, dans un deuxième temps, à faire une interprétation des résultats de cette analyse. Parmi les travaux effectués dans ce domaine, citons en particulier ceux de SALTON (Etats-Unis [27][28]). L'Institut de la Langue Française, dans ses travaux de lexicométrie (M. TOURNIER [25]) a également étudié les liens sémantiques existant entre les termes de notre langue : tous les couples de termes possibles sont étudiés. Le principe est basé également sur la co-occurrence des termes. L'analyse tient compte des distances entre chacun des éléments du couple à étudier (c'est-à-dire des nombres de termes qui séparent les occurrences des termes du couple considéré). Les liens sont ensuite interprétés pour les couples dont les éléments apparaissent ensemble avec un taux de fréquence plus élevé qu'un certain seuil paramétré [25].

On peut aisément prévoir que plus un réseau sémantique est dense, plus le taux de rappel[1] est élevé.
Le rappel est également meilleur quand les plausibilités de proximité des objectifs sont plus élevées ou quand les restrictions décrites dans la partie des liens sémantiques (cf. domaine de validité p. 64) sont plus réduites.

---

(1) Le rappel et la précision sont deux ratios grâce auxquels on peut évaluer un système documentaire :

$$\text{rappel} = \frac{\text{Nombre de documents pertinents proposés}}{\text{Nombre de documents pertinents dans la collection}}$$

$$\text{précision} = \frac{\text{Nombre de documents pertinents proposés}}{\text{Nombre de documents proposés}}$$

Un système documentaire idéal est celui dans lequel, pour une requête donnée, tous les documents pertinents de la collection sont proposés (Rappel = 1) et tous les documents proposés sont pertinents (précision = 1).

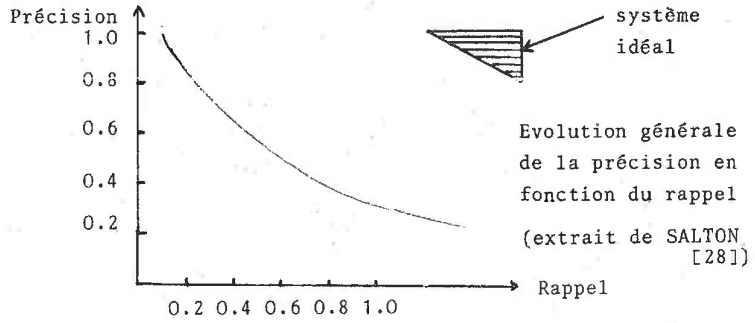
La pertinence des documents d'une collection par rapport à une requête est déterminée par un groupe d'experts.

La construction du thésaurus doit être réalisée en fonction du rappel et de la précision[1] souhaités par les utilisateurs. Leurs besoins étant souvent assez mal connus initialement, leurs appréciations pourront servir à l'ajustement du système afin qu'il soit progressivement mieux adapté à leurs besoins.

La qualité des services d'un documentaliste varie suivant ses connaissances sur le sujet des requêtes qu'il prend en compte. De la même façon, la pertinence réelle des documents proposés par le système dépend de la qualité et de la façon d'utiliser les connaissances qu'on lui fournit.

---

(1) Le rappel varie généralement inversement par rapport à la précision : si on veut poursuivre la recherche pour trouver davantage de documents pertinents, la précision baisse car on trouve aussi beaucoup de documents non pertinents.
Le lecteur doit donc choisir entre obtenir un fort rappel et une précision faible ou une haute précision et un rappel faible.



Evolution générale de la précision en fonction du rappel

(extrait de SALTON [28])

3.2.2. Description de l'amplification de la portée d'une requête grâce au thésaurus

Rappelons que l'amplification a pour but d'associer aux objectifs qui traduisent une requête, des objectifs qui leur sont plus ou moins proches sémantiquement.

Cette amplification est nécessaire pour pouvoir mener des recherches exhaustives et sélectionner des documents qui ne sont pas indexés par les objectifs qui traduisent les requêtes, mais par d' autres objectifs qui leur sont voisins.

3.2.2.1. Principe général

Nous avons vu comment était associé un objectif à un autre qui lui est relié directement. Certains objectifs sont reliés indirectement à un objectif appartenant à une requête.

Pour tout objectif associé, il convient d'avoir sa relation et sa plausibilité de proximité définissant son lien avec l'objectif initial pour lequel on cherche des objectifs à associer. Il est donc nécessaire d'établir des règles de composition des relations et des plausibilités. Il sera possible ainsi de maîtriser une amplification et de l'arrêter quand les objectifs qui pourraient être encore associés ne sont plus suffisamment en rapport avec la requête présentée.

Par ailleurs, pour associer des objectifs à un objectif appartenant à une requête donnée, il faut tester, par

rapport aux objectifs de cette requête, la similarité des domaines de validité des liens sémantiques concernés par les associations.

Nous allons à présent décrire plus précisément comment sont effectuées les associations.

3.2.2.2. Composition des relations

Soient obj_o un objectif d'une requête

obj_i le $i^{ième}$ objectif qui lui est associé

obj_k un objectif directement relié à obj_i ((obj_i, obj_k) rel pl) et pouvant être éventuellement le $k^{ième}$ objectif associé à obj_o :

$$rel(obj_o, obj_k) = rel(obj_o, obj_i) \circ rel(obj_i, obj_k)$$

Les relations sont composées à l'aide du tableau de nombres entiers COMP.

Chaque relation y sera représentée par son numéro d'ordre.

VD	=	1
VDI	=	2
SD	=	3
SD-1	=	4
PD	=	5
PD-1	=	6
MOYEN	=	7
MOY-1	=	8
ASS	=	9
ASDI	=	10

Tableau de composition des relations :

REL 1 =	REL 2 = 1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	2	3	4	5	6	7	8	9	10
3	3	3	3	9	5	6	7	8	9	10
4	4	4	9	4	5	6	7	8	9	10
5	5	5	5	5	5	9	7	8	9	10
6	6	6	6	6	9	6	7	8	9	10
7	7	7	7	7	7	7	7	9	9	10
8	8	8	8	8	8	8	9	8	9	10
9	9	9	9	9	9	9	9	9	9	10
10	10	10	10	10	10	10	10	10	10	10

On peut déduire de ce tableau que l'ordre de priorité des relations est approximativement le suivant : ass et asdi, moyen et $moyen^{-1}$, pd et pd^{-1}, sd et sd^{-1}, vd et vdi.

Exemple 1 :

(automobile, voiture) sd 0.9

(Certaines voitures ne sont pas des automobiles (ex : voitures à traction animale)

(volant, automobile) pd 0.8

rel(volant, voiture) = rel(voiture,automobile) o rel(automobile,volant)

= sd o pd = pd

Exemple 2 :

(automatisation,activité) sd 0.95

(méthode,activité) moyen 0.95

(corig,méthode) sd 0.9/automatisation

(corigB,corig) pd 0.9

Rel(automatisation,méthode) =

Rel(automatisation,activité) o rel(activité,méthode) =

moyen o sd = moyen

Rel(automatisation,corig) =

Rel(automatisation,méthode) o rel(méthode,corig) =

moyen o sd = moyen

Rel(automatisation,corigB) =

Rel(automatisation,Corig) o rel(corig,corigB) =

moyen o pd = moyen

Récapitulatif des relations d'objectifs à l'objectif "automatisation" :

Objectifs	Relations
automatisation	vd
activité	sd^{-1}
méthode	moyen
corig	moyen
corigB	moyen

Dans le système $Satin_1$ et Mistral, les compositions autorisées sont explicitées par l'utilisateur. Dans la plupart des autres systèmes, la composition des relations est permise mais l'impossibilité d'effectuer des appauvrissements (cf. chapitre à la page suivante) contraint les auteurs à limiter le nombre des compositions en série.

3.2.2.3. Calcul de la proximité de deux objectifs reliés indirectement

Soient un objectif obj_o appartenant à une requête
un objectif obj_i qui lui est associé
un objectif obj_k directement relié à obj_i ((obj_i, obj_k) rel pl)
et qu'on voudrait associer indirectement à obj_o

$$plprox(obj_o, obj_k) = plprox(obj_o, obj_i) \ S \ plprox(obj_i, obj_k)$$

Exemple 1 :

(automobile,voiture) sd 0.9

⇒ plprox(automobile,voiture) = 0.8 S 0.9 = 0.75

(volant,automobile) pd 0.8

⇒ plprox(volant,automobile) = 0.8 S 0.8 = 0.7

plprox(volant,voiture) = plprox(voiture,automobile) S plprox(automobile,volant) = 0.75 S 0.7 = 0.62

Exemple 2 :

(automatisation,activité) sd 0.95

(méthode,activité) moyen 0.95

(corig,méthode) sd 0.9/automatisation

(corigB,corig) pd 0.9

Association d'objectifs à l'objectif "automatisation" :

Objectifs	Relations	Plausibilités de proximité
automatisation	vd	1
activité	sd^{-1}	0.9 S 0.95 = 0.87
méthode	moyen	0.87 S 0.9 S 0.95 = 0.78
corig	moyen	0.78 S 0.8 S 0.9 = 0.66
corigB	moyen	0.66 S 0.8 S 0.9 = 0.60

Un seuil de proximité définit le seuil limite de proximité en dessous duquel les associations ne sont pas autorisées; Par conséquent, si obj_o est un objectif qui indexe une requête, tout objectif dont la plausibilité de proximité par rapport à obj_o est inférieure au seuil de proximité ne peut pas être inclus dans la liste des objectifs qui représentent une requête. Ceci constitue la première condition d'une association.

La deuxième est que le domaine de validité du lien sémantique qui sert à l'association éventuelle d'un objectif doit se rapporter suffisamment à la requête présentée ou être absent.

3.2.2.4. Test de la similarité d'un domaine de validité par rapport à une requête

Il n'est pas toujours immédiatement évident, pour le système documentaire, de détecter le rapport existant entre un domaine de validité et une requête. Pour découvrir ce rapport éventuel, il est donc souvent nécessaire d'amplifier la portée d'un domaine de validité.

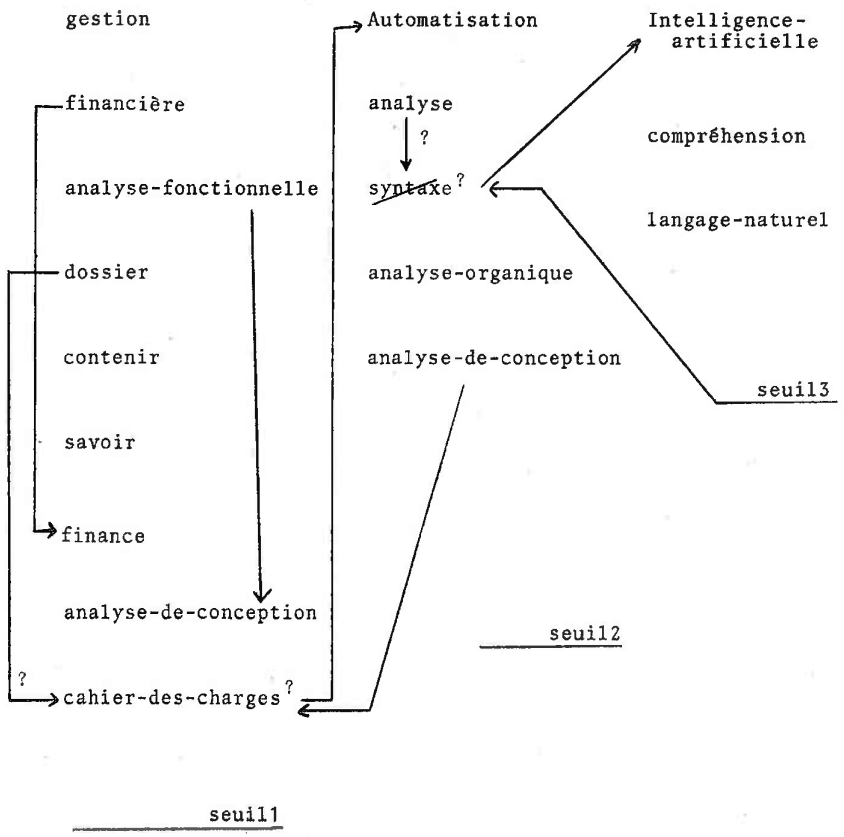
Exemple :

requête : "J'aimerais savoir ce que doit contenir un dossier d'analyse fonctionnelle portant sur de la gestion financière"

thésaurus :

(syntaxe,analyse) moyen 0.9 / intelligence artificielle
(compréhension,intelligence artificielle) ass 0.8
(finance,comptabilité) ass 0.95
(cahier-des-charges,dossier) sd 0.65 / automatisation
(analyse-fonctionnelle,analyse-de-conception) vd 0.95
(analyse organique,analyse) sd 0.8 / gestion
(analyse-de-conception,analyse) sd 0.8 / gestion
(analyse,automatisation) moyen 0.9 / gestion
(financière, finance) ass 1

Amplification de la portée de la requête :



La première liste (à gauche) est la liste des objectifs représentant la requête : les six premiers la traduisent initialement.

Deux objectifs peuvent être ajoutés à l'aide de deux liens sémantiques ne comportant pas de domaine de validité :

- "finance" à "financière"
- "analyse-de-conception" à "analyse-fonctionnelle".

L'association suivante nécessite une vérification du contexte : Le système cherche à savoir s'il peut enrichir la requête à l'aide de l'objectif "cahier-des-charges".
Le lien sémantique concerné est :

(cahier-des-charges, dossier) sd 0.65 / automatisation

Pour que l'association puisse se faire, il faut que le domaine de validité "automatisation" soit suffisamment en rapport avec la requête. Pour tester ce rapport, le système interrompt momentanément l'amplification de la requête et cherche à amplifier la portée de "automatisation" (2ème liste). Une jonction s'établissant entre les 2 listes, l'association est autorisée.

Mais, le même problème d'association conditionnelle, causé par la présence d'un domaine de validité s'était encore posé lors de l'amplification de la portée de "automatisation" : dans la 2ème liste, le système doit en effet s'interroger sur la possibilité d'associer indirectement "syntaxe" à "automatisation".

Le lien sémantique concerné est (syntaxe,analyse) moyen 0.9/ Intelligence Artificielle. En amplifiant le domaine de validité "Intelligence Artificielle" (3ème liste), le système constate que cet objectif ne se rapporte pas à "automatisation" ou à la requête. L'association envisagée n'est donc pas admise.

Description générale du test de similarité d'un domaine de validité par rapport à une requête

Le système autorise, au maximum, trois domaines de validité en cours d'amplification, mais les seuils au dessus desquels des associations sont autorisées, sont de plus en plus élevés, au fur et à mesure qu'on "s'écarte" de l'amplification de la requête.

Exemples : seuil1(amplification requête) : 0.55
seuil2(amplification 1er domaine de validité) : 0.65
seuil3 (amplification 2ème domaine de validité) : 0.75
seuil4 (amplification 3ème domaine de validité) : 0.85

Soit $i \in \{1,2,3,4\}$ le n° des listes d'objectifs en cours d'amplification.
Soit obj_k ($k \in \{1,2,3\}$) le dernier objectif de la liste k (celui dont le test d'association a provoqué l'amplification de la liste k+1).
Pour $i > 1$, un objectif est associé dans la $liste_i$
- si sa plausibilité de proximité par rapport à l'objectif qui est à la tête de la $liste_i$ est supérieure au $seuil_i$ et
- si le domaine de validité de l'association est inexistant ou égal à l'un des objectifs de la $liste_i$

L'amplification d'une liste i ($i > 1$) est arrêtée
- si un objectif associé dans la $liste_i$ appartient à la $liste_{i-1}$. Dans ce cas, obj_{i-1} est associé
- si les associations dans la $liste_i$ ne sont plus possibles, le seuil d'amplification limite ($seuil_i$) étant dépassé. Dans ce cas, l'association de obj_{i-1} n'est pas autorisée.

Stopper l'amplification d'une $liste_i$ ($i > 1$) provoque la poursuite de l'amplification de la $liste_{i-1}$, exactement là où elle s'était arrêtée. L'ordre LIFO est donc suivi. On peut par conséquent utiliser une pile, de hauteur maximale : 4.
Dans chaque élément est indiquée, en particulier, l'adresse dans le thésaurus à partir de laquelle l'exploration doit se poursuivre.

Exemple :

Dans l'exemple précédent, lors de la recherche d'objectifs à associer à "compréhension" ($liste_3$), l'état de la pile était le suivant :

pointeur tête de pile →

Intelligence-artificielle	compréhension	(2,1)	adf_3
automatisation	analyse	(1,2)	adf_2
gestion	dossier	(4,2)	adf_1
nom de la liste à développer	objectif pour lequel on cherche des objectifs à associer	adresse de l'exploration en cours dans le thésaurus (ligne,colonne)	adresse de fin de la classe du thésaurus à explorer (1)

(1) Pour accélérer les recherches, le thésaurus est décomposé en classes (cf. chapitre "subdivision du thésaurus en classes, p. 157).

3.2.3. Conclusion : les limites d'un thésaurus

Les liens sémantiques à mettre dans un thésaurus doivent représenter des rapports qui sont implicites pour chacun de nous.

Exemples :

(programmation,activité) sd 0.95
(méthode,activité) moyen 0.95
(date,précis) ass 0.95
...

De telles connaissances permettent de doter le système d'une certaine intelligence. Elles améliorent donc la compréhension des requêtes et, par conséquent, la qualité des recherches documentaires.

Par souci d'homogénéité, tous les liens sémantiques ont la même forme. Une forme variable, permettant par exemple l'indication de plusieurs conditions d'association serait cependant mieux adaptée pour exprimer certains rapports entre objectifs.

Par ailleurs, le thésaurus décrit est peu structuré. Les regroupements des connaissances apparaissent mal.

Nous allons décrire, à présent, un outil permettant un autre type d'enrichissement des requêtes que le thésaurus : il est constitué par des procédures, appelées "procédures stratégiques".

3.3. Les procédures stratégiques

3.3.1. Définition et rôle

Les procédures stratégiques sont constituées de connaissances générales ou concernant les utilisateurs. Elles ont pour but l'enrichissement des requêtes grâce à l'évaluation de l'esprit dans lequel les recherches documentaires doivent être effectuées. Elles servent donc à préciser le cadre du sujet informatique des requêtes.

Exemple :

Soit, par exemple, l'objectif informatique "fichier". Si le lecteur est un chef de projet, les livres se rapportant à ce sujet et correspondant à ses besoins ne sont généralement pas les mêmes que ceux qui sont à conseiller à un pupitreur qui s'intéresse plus à la manipulation des fichiers qu'à leur conception.

Il n'est évidemment pas interdit à un opérateur de vouloir savoir comment se passent les choses à un niveau qui ne le concerne pas directement. Mais, nous supposons que si un documentaliste sait qu'un opérateur lui demande un livre sur les fichiers et si deux livres sur les fichiers figurent dans les rayons, l'un sur la manipulation des fichiers, l'autre sur la conception des fichiers, le documentaliste aura a priori une légère préférence pour lui conseiller le premier livre. Il ne lui proposera le second que sur demande plus explicite.

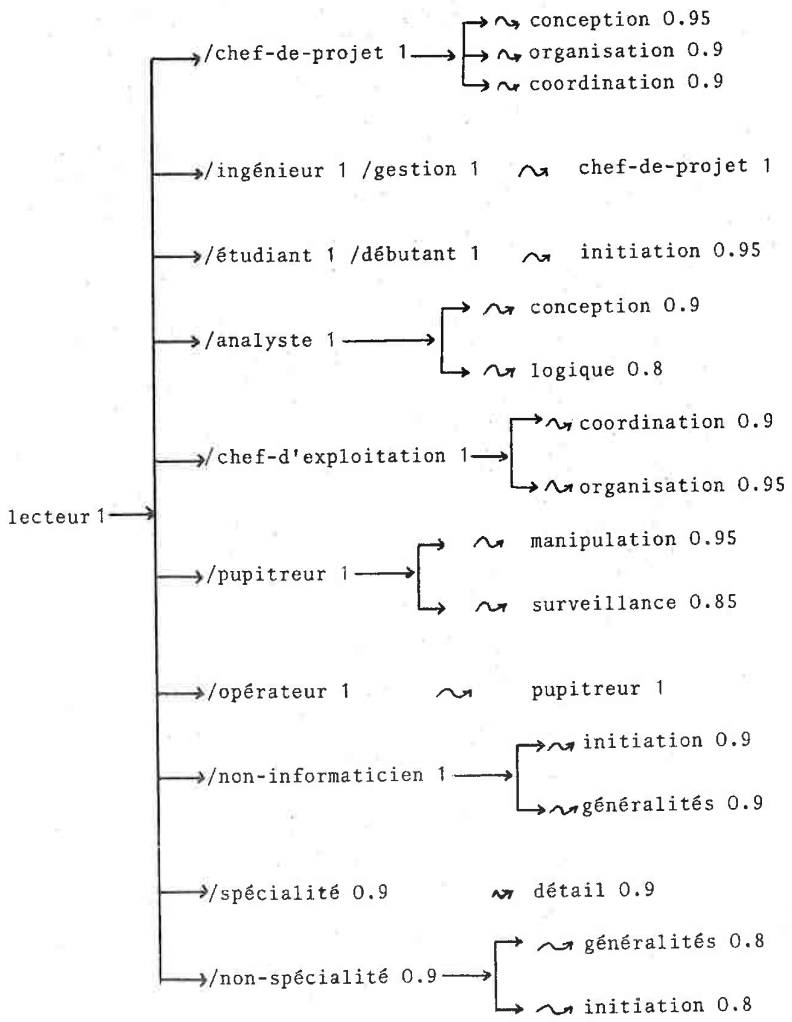
Cet exemple montre l'intérêt que pourrait apporter un dialogue entre le système documentaire et l'utilisateur, dans certains cas d'ambiguïté, notamment quand une requête est imprécise ou quand elle peut induire en erreur.

Dans une requête, les adverbes interrogatifs tels que "où", "quand", "pourquoi", "comment" ... et les objectifs tels que "vouloir connaître", "choisir", "estimer", "se renseigner"... précisent également le contexte ou la nature de la requête présentée.

Certains objectifs appartenant aux connaissances générales ou relatives aux lecteurs permettent un enrichissement meilleur que les autres car leur contribution à l'orientation de la recherche vers les besoins réels exprimés par les lecteurs est plus large. Ces objectifs vont désigner des procédures stratégiques.

3.3.2. Description et exploration des procédures stratégiques

Exemple : procédure stratégique "lecteur" (dans le cadre d'une bibliothèque d'informatique)



Une procédure a une structure arborescente.

La racine du graphe est le nom de la procédure et son point d'entrée unique.

est le symbole de disjonction.

Les questions d'une procédure

Toute procédure est un questionnaire : tout objectif figurant après un slash "/" est une question à laquelle le système est chargé de répondre.

Une plausibilité, appelée "plausibilité d'importance" figure après chaque question. Par convention, plus la plausibilité d'importance d'une question tend vers 1, plus cette question est importante.

Un chemin est une suite de questions apparaissant consécutivement dans une procédure.

Exemples de chemin :

- lecteur / ingénieur / gestion
- lecteur / pupitreur

Enrichissement d'une requête grâce aux procédures stratégiques

Lorsque l'exploration atteint une feuille introduite par le symbole "$\sim$", l'objectif obj qui représente cette feuille est associé aux objectifs de la requête. Les questions successives situées sur le chemin allant de la racine de la procédure à obj représentent les conditions à remplir pour enrichir la requête grâce à l'objectif obj.

Tout comme avec les liens sémantiques, les objectifs sont plus ou moins intéressants à associer. L'enrichissement de

la requête par un objectif obj varie en fonction de la plausibilité d'intérêt attachée à obj. La relation "associé à" (ass) est employée pour ce type d'association.

Exploration d'une procédure

Le but de l'exploration d'une procédure est d'atteindre les feuilles de l'arborescence.

Si la réponse à une question Q est affirmative, le système prend ensuite en considération les successeurs de Q. Le graphe est exploré en fonction de l'estimation des noeuds déjà atteints (cf. chapitre : "ordre de priorité des questions", p.125).

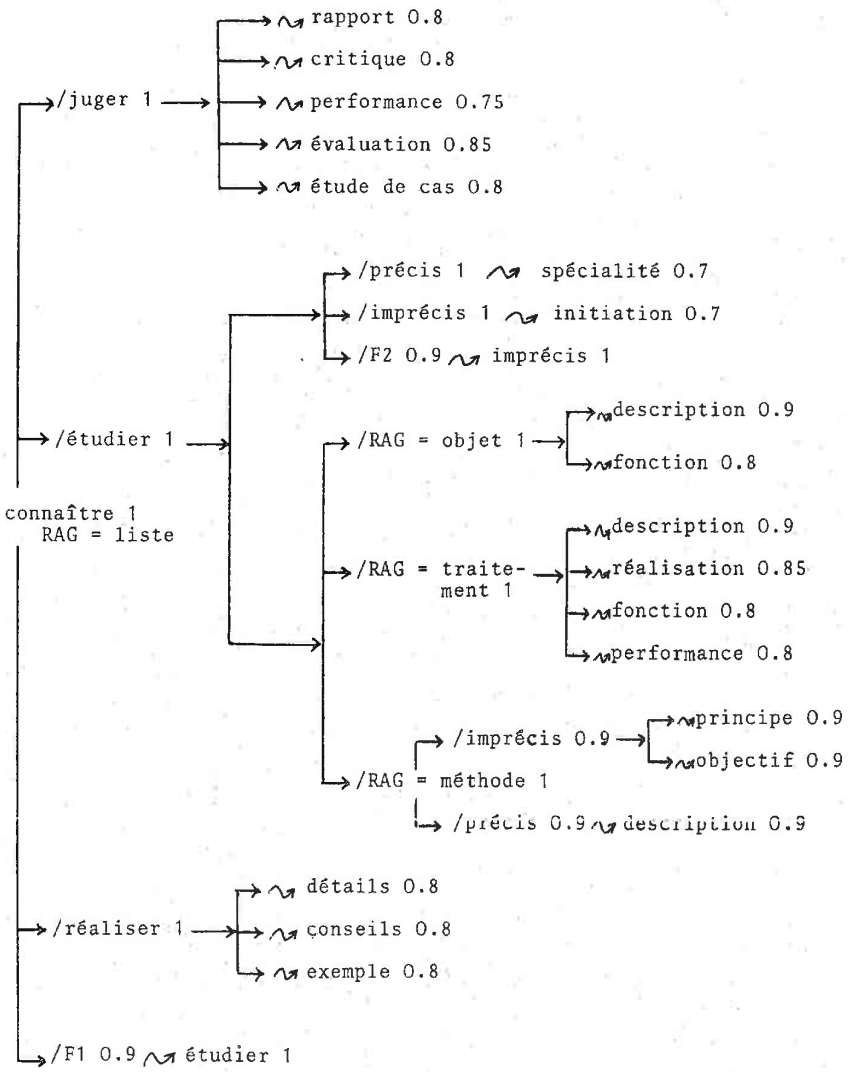
Détermination des réponses aux questions

Si Q est une question représentée par un objectif, la réponse à Q est affirmative si l'objectif Q est suffisamment en rapport avec la requête.

Après une recherche se soldant par un insuccès, si la réponse à une question Q est indéterminée et si la plausibilité d'importance de Q est faible (<seuil-plimp), le système passe directement aux successeurs de Q. Le seuil seuil-plimp est initialisé à 0.5 et croît au fur et à mesure de la recherche: en cas d'insatisfaction prolongée de la requête, le système ne s'attarde donc que sur les questions les plus importantes.

Il évite ainsi de bloquer la recherche (cf. exemple p. 128).

Procédure stratégique "connaître"



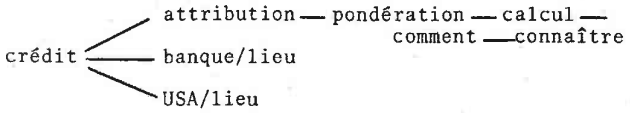
Interprétation de la procédure stratégique "connaître" :

RAG (Reconnaissance Antécédent Gauche) indique la liste des objectifs concernés directement par "connaître".

Exemple :

requête : "Je désirerais connaître comment on calcule les pondérations d'attribution de crédit dans une banque aux USA"

traduction :



RAG est, dans cet exemple, la liste d'objectifs allant de "connaître" à "crédit" (le lecteur ne désire pas vraiment connaître les banques aux USA).

Le sens de "connaître" varie suivant son contexte : on peut vouloir connaître des informations pour juger, pour étudier ou pour réaliser quelque chose.

Le choix entre ces trois objectifs n'est pas toujours évident. Il convient cependant de ne pas bloquer trop longtemps les recherches.

En cas d'insatisfaction prolongée de la requête, le procédé est le suivant : F1 (au bas de la page 87) est un objectif dénué de sens. Nous lui avons associé une plausibilité d'importance assez faible afin qu'il puisse être "outrepassé"(1) après plusieurs tentatives infructueuses. Par ce biais le système impose un objectif type à "connaître" qui est "étudier" bien que ce terme ne soit pas apparu lors des questions précédentes "juger", "étudier" et "réaliser".

Si le lecteur veut connaître pour juger, il est intéressant de lui fournir des rapports (par opposition à des comptes-rendus) ou des livres dans lesquels il y a des critiques, des performances, des évaluations, des études de cas ; si les conditions "connaître" et "juger" sont remplies, la requête peut donc être enrichie des cinq objectifs "rapport", "critique ", "performance ", "évaluation ", "étude-de-cas".

Si le lecteur veut connaître pour étudier :

- Si c'est précis, le système supposera que la requête se rapporte au domaine de spécialité du lecteur.
- Si c'est imprécis (exemples : "je désirerais connaître la méthode Warnier" ou "je voudrais connaître le PL/I"), il s'agit probablement d'un lecteur qui veut s'initier (au Warnier ou au PL/I).

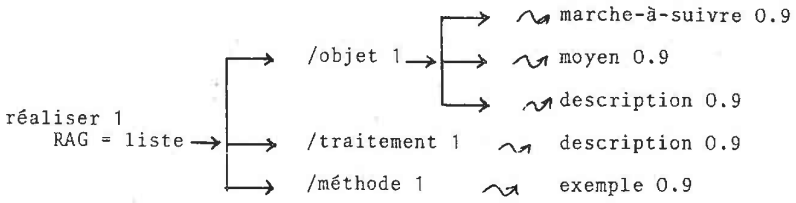
A défaut de pouvoir choisir entre "précis" et "imprécis", c'est l'objectif "imprécis" qui sera sélectionné par le même procédé que celui qui a été décrit plus haut.

(1) L'exemple, page 128 , montre comment le système peut, dans certains cas, forcer un passage à un niveau plus profond de la procédure.

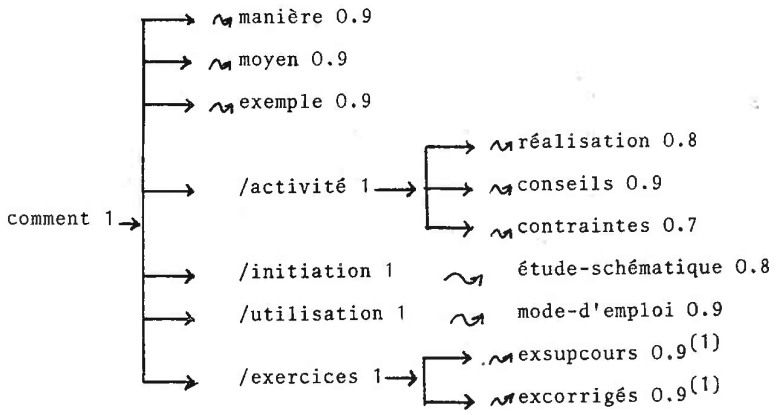
- Si le lecteur veut étudier un objet, les objectifs "description", et "fonction" de l'objet sont considérés sous-entendus dans la requête.
- S'il s'agit de l'étude d'un traitement, le lecteur veut probablement connaître sa description, sa réalisation, ses fonctions, et ses performances.
- S'il veut étudier pour connaître une méthode et si c'est imprécis (exemple : je veux connaître le corig), il faut qu'il sache avant tout le principe et les objectifs de cette méthode.
- Si c'est précis (exemple "je désirerais savoir comment on construit un schéma directeur et un schéma de circulation des informations (en corig), il faut essayer de trouver des manuels dans lesquels il y a une description de la méthode.

Si le lecteur veut connaître pour réaliser quelque chose, il lui faut des détails, des conseils pour la réalisation et des exemples de ce qu'il veut réaliser.

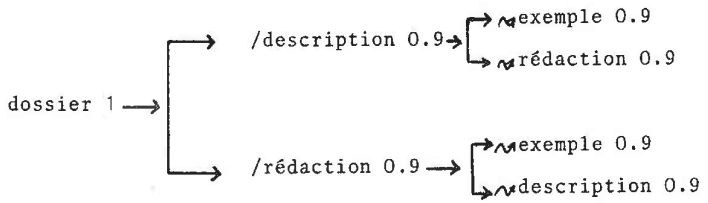
Procédure stratégique "réaliser"



Procédure stratégique "comment"



Procédure stratégique "dossier"



---

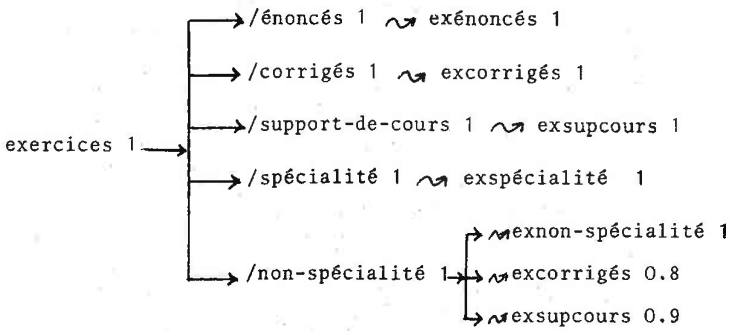
(1) Des objectifs composés peuvent être construits

exemples :

exsupcours = exercices avec support de cours

excorrigés = exercices avec leur corrigé

Procédure stratégique "exercices"



3.4. Conclusion

En résumé, les étapes de l'analyse d'une requête sont donc les suivantes :

- filtrage de ses objectifs
- analyse syntaxique de cette requête afin d'ordonner ses objectifs
- amplification de la portée de cette requête, par l'intermédiaire
  - . du thésaurus, qui doit contenir les liens sémantiques implicites pour chacun de nous
  - . des procédures stratégiques destinées à enrichir la requête en évaluant l'esprit dans lequel la recherche documentaire doit être effectuée.

La recherche documentaire consiste en une confrontation des requêtes analysées avec les informations documentaires. Il convient donc, à présent, d'étudier la représentation du fonds documentaire avant de pouvoir établir cette comparaison entre requêtes et documents.

IV. LA MODELISATION

DU FONDS DOCUMENTAIRE

IV. LA MODELISATION DU FONDS DOCUMENTAIRE

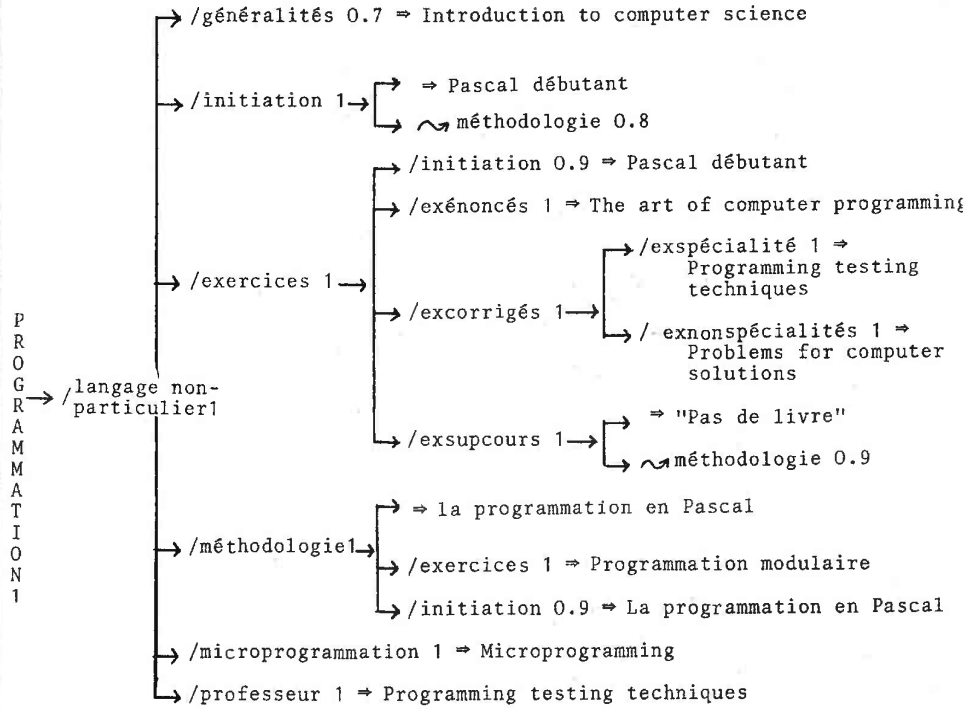
Le fonds documentaire est représenté à l'aide de procédures documentaires.

Description et exploration des procédures documentaires :

Une procédure documentaire est un questionnaire qui porte sur un domaine de la matière documentaire et qui permet la sélection de documents pertinents qui appartiennent à ce domaine.

Pour explorer les connaissances documentaires et répondre aux questions, le système procède de la même façon qu'avec les connaissances stratégiques.

Exemple : Procédure documentaire "programmation"



Cette procédure documentaire est un exemple simplifié : nous avons voulu y faire apparaître les principaux types d'informations (décrits à la page suivante) ainsi que les différentes sortes d'objectifs pouvant servir à la constitution des chemins d'accès aux livres :

- objectifs informatiques (ex : microprogrammation)
- objectifs exprimant des types de lecteur (ex : professeur)
- objectifs généraux (ex : généralités, initiation, exsupcours (= exercices avec support de cours)...).

1. DESCRIPTION DES PROCÉDURES DOCUMENTAIRES

Les procédures documentaires peuvent être explorées en même temps que les procédures stratégiques, bien qu'elles jouent un rôle souvent différent (recherche de documents au lieu d'amplification des requêtes). Les mécanismes utilisés sont semblables : les procédures documentaires constituent aussi un questionnaire en forme d'arborescence dont les noeuds sont des questions et dont les feuilles sont des actions.

- Si une feuille atteinte est précédé du symbole "$\rightsquigarrow$", elle indique un objectif qui enrichit la requête.

La frontière entre procédures stratégiques et procédures documentaires n'est donc pas très rigoureuse.

- Si une feuille est entourée de guillemets précédés du symbole " $\Rightarrow$", il s'agit d'un message à envoyer aux lecteurs.

Ex : $\Rightarrow$ "pas de livre".

- Si seul le symbole "$\Rightarrow$" précède une feuille, cette dernière représente dans ce cas un pointeur qui indique l'adresse des données signalétiques d'un livre(1).

Les objectifs allant de la référence d'un livre L au nom de la procédure (inclus) indexent L.

---

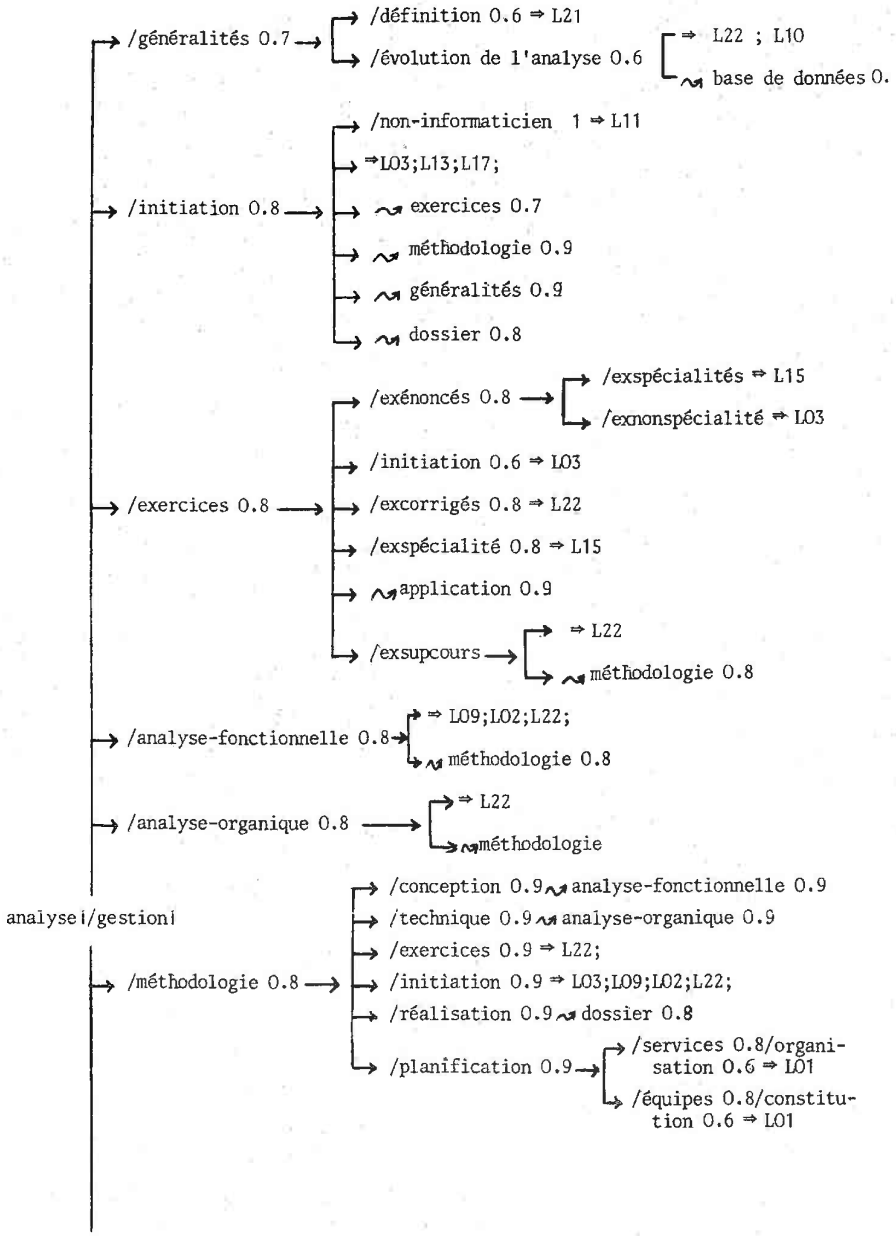
(1) les données signalétiques (ou descriptions bibliographiques) forment les caractéristiques formelles des documents, c'est-à-dire les informations qui, sans toucher à leur contenu, permettent de les identifier (titre, auteur, source...) et de les enregistrer sur une notice bibliographique.

Exemples :

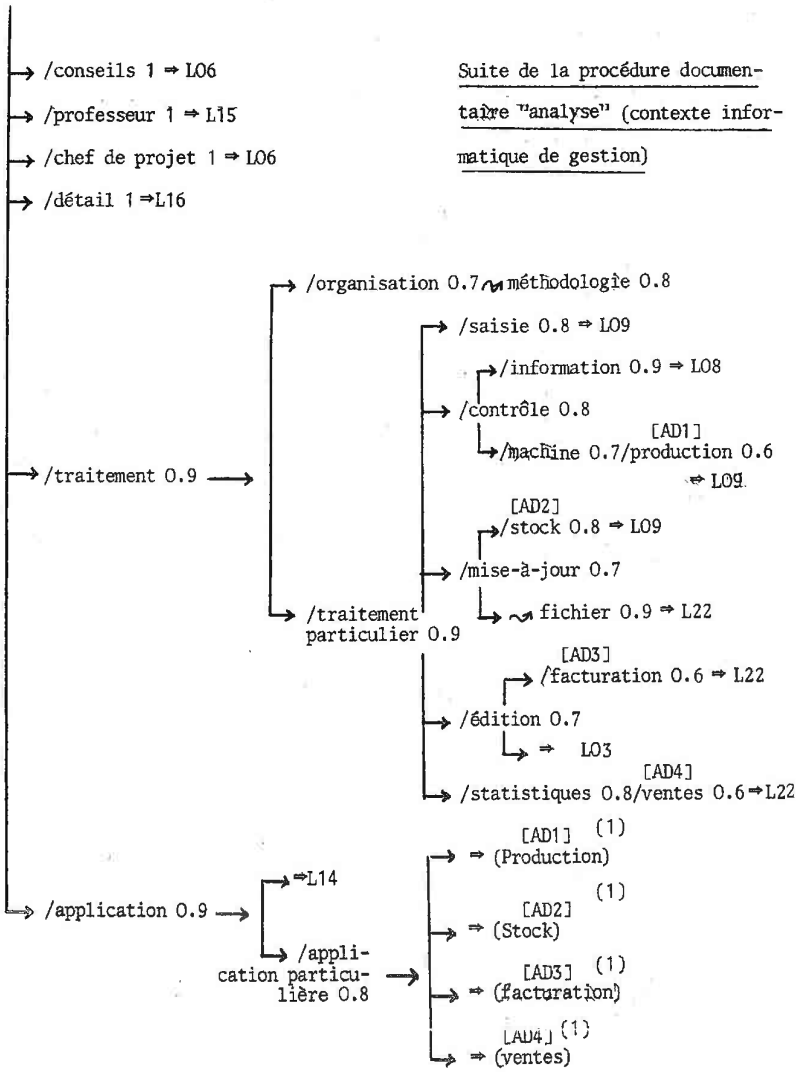
- Le livre intitulé "the art of computer programming" est indexé par les objectifs "programmation", "langage non particulier", "exercices" et "exénoncés".
- "Pascal débutant" désigne le titre d'un livre pertinent pour le chemin ayant conduit à ce point (programmation/langage non-particulier/initiation). Le lecteur n'ayant pas émis l'intérêt pour un langage particulier, dans sa requête, le système lui conseille de programmer en Pascal, par exemple .

Pour essayer d'atteindre un livre L, le système doit s'interroger sur la similarité entre les objectifs qui indexent L et les objectifs qui représentent la requête. Si L est atteint, L est pertinent puisque les objectifs qui l'indexent ont un rapport suffisamment étroit avec la requête.

Procédure documentaire "analyse"

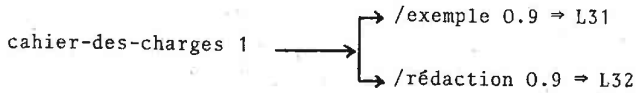


Suite de la procédure documentaire "analyse" (contexte informatique de gestion)

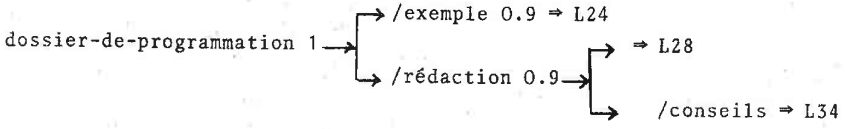


(1) Ces renvois internes ont pour but de contourner l'obstacle représenté par "traitement particulier" : des livres décrivant des applications particulières pourront en effet être atteints sans que le lecteur ait à demander des traitements particuliers sur ces applications particulières.

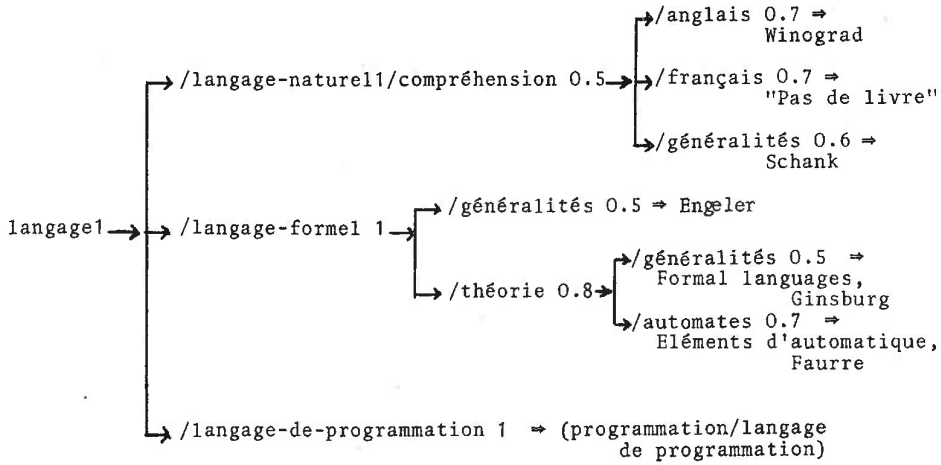
Procédure documentaire "cahier des charges"



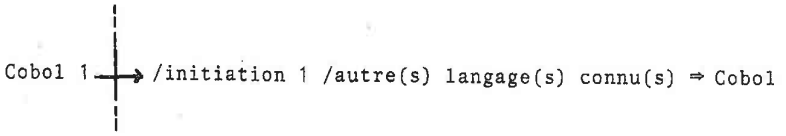
Procédure documentaire "dossier de programmation"



Procédure documentaire "langage"



Procédure documentaire "Cobol"



Pour choisir les documents les plus pertinents possibles, il est intéressant de savoir si le lecteur connaît déjà d'autres langages de programmation que Cobol.

2. Mode de construction des procédures documentaires et stratégiques

Le coût et le temps nécessaires à la constitution d'une procédure stratégique ou documentaire sont difficiles à évaluer car nous nous sommes limités à quelques procédures expérimentales en vue d'illustrer le fonctionnement du système. Néanmoins, il est possible de faire certaines remarques sur la façon dont nous nous y sommes pris pour les écrire.

Rappelons qu'une procédure stratégique doit permettre d'associer des objectifs nouveaux à ceux qui se trouvent dans la requête, tandis qu'une procédure documentaire doit à travers ces objectifs conduire à des références d'ouvrages.

De façon générale, il est plus simple de concevoir des procédures courtes. Il ne faut pas pour autant caricaturer la réalité en faisant l'impasse d'un certain nombre de critères. En particulier une procédure documentaire ne se réduit pas à une simple classification des domaines. La démarche à suivre peut consister pour chaque ouvrage à répondre à la question : qui peut être intéressé par l'ouvrage et pourquoi ? (ex : à titre d'entrainement ? ; de documentation ? ...). Il reste, et ceci est délicat faute de méthodologie, à regrouper les réponses récoltées pour les différents livres. On ne rencontre pas cette difficulté avec les procédures stratégiques.

D'un point de vue plus pratique, il apparaît que des faibles plausibilités associées aux questions d'une procédure documentaire ne font pas suffisamment obstacle car elles sont très

rapidement forcées (cf. chapitre "détermination de réponses aux questions", page 86). Il convient donc d'en réduire le nombre et de les répartir judicieusement afin qu'elles remplissent le rôle de "soupape de sécurité" (éviter les blocages).

Si trop de "déblocages" sont effectués pour trouver les documents, l'estimation de ceux-ci n'est plus suffisamment précise. Ceci nous incite donc aussi à limiter la profondeur des procédures. Il ne faut pas tomber dans l'excès inverse. La profondeur optimale croît avec la richesse du fonds. Dans nos essais elle est de l'ordre de 3 à 4.

Conclusion

Le lexique, le thésaurus, les procédures stratégiques et documentaires constituent les connaissances que le système documentaire va utiliser pour réaliser des recherches documentaires.

Nous allons décrire à présent comment sont effectués les traitements correspondant à la recherche documentaire.

V. LA RECHERCHE

DOCUMENTAIRE

V. LA RECHERCHE DOCUMENTAIRE

1. INTRODUCTION

Les entrées du module "recherche documentaire" sont constituées des connaissances du système (lexique, thésaurus, procédures stratégiques et documentaires, requêtes présentées). Ces entrées ayant été décrites, nous allons analyser les sorties du module puis les règles qui permettront de les obtenir à partir des entrées.

Nous décrirons ensuite les conditions d'arrêt d'une recherche documentaire, la stratégie de recherche suivie, puis nous terminerons en traitant complètement quelques exemples de requêtes. Avant tout, nous allons exposer les particularités principales de la recherche documentaire effectuée par notre système.

2. PARTICULARITÉS DE LA RECHERCHE DOCUMENTAIRE EFFECTUÉE PAR NOTRE SYSTEME

Pour chaque requête, nous voulons éditer un état sur lequel nous voulons faire apparaître, grâce au réseau des approximations (thésaurus) et aux procédures stratégiques, une mesure de la pertinence de chaque livre proposé par rapport à la requête, mais surtout aussi, d'une façon plus détaillée, par rapport à chacun des objectifs de cette requête(1).

---

(1) Une telle analyse, aussi détaillée, nous paraît être originale : Dans le système Smart [28], par exemple, la pertinence d'un livre est exprimée globalement en fonction d'une requête : Rappelons que chaque document et chaque requête sont d'abord transformés en un vecteur composé d'un ensemble de couples dont le premier élément est un nombre correspondant à un descripteur et le deuxième, un nombre indiquant le poids ou l'importance relative du descripteur dans le document ou la requête.
Le fonds documentaire pourrait donc être représenté par une matrice : pour un concept$_i$ et un document$_j$, l'élément de la matrice C_j^i représenterait l'importance du concept$_i$ dans le document$_j$.
Le système Smart mesure la corrélation entre le vecteur D_j d'un livre et celui d'une requête globale R_i par l'intermédiaire de la fonction Cosinus :

$$\text{Cosinus}(R_i, D_j) = \frac{\sum_{k=1}^{m} R_k^i \, D_k^j}{\sum_{k=1}^{m} (R_k^i)^2 \times \sum_{k=1}^{m} (D_k^j)^2}$$

m est le nombre de concepts dans le corpus
R_k^i est le poids du $concept_k$ dans la requête R_i, D_k^j est le poids du $concept_k$ dans le document D^j. Les valeurs de la fonction appartiennent à l'intervalle [0,1] (1 pour une corrélation parfaite).

Le lecteur pourra ainsi filtrer les documents qui décrivent suffisamment les objectifs les plus importants de sa requête.

Cette analyse très fine de la pertinence des documents pourrait remplacer dans la plupart des systèmes documentaires actuels les résumés encombrants qui figurent après les références des livres proposés.

Par ailleurs, nous ne voulons pas que le système effectue complètement l'analyse d'une requête avant de faire une recherche documentaire :
L'amplification des requêtes ne sera effectuée qu'en fonction des succès obtenus par la recherche documentaire, qui sera elle-même guidée par l'estimation des noeuds atteints.

La recherche avec notre système consiste à "brancher" les procédures documentaires et stratégiques concernées par une requête puis à explorer ces procédures sélectionnées(1).

---

(1) On peut remarquer que le système Smart effectue également une recherche en deux approches :
Voici comment se déroule cette recherche : le fonds documentaire est auparavant réparti en sous-ensembles ("clusters"). Un sous-ensemble est une collection de documents suffisamment similaires pour les regrouper en une même classe. Pour chaque sous-ensemble, un document appelé "vecteur central" est choisi pour représenter l'ensemble des documents du sous-ensemble.
Un vecteur requête est d'abord comparé à chacun des vecteurs centraux représentatifs des sous-ensembles.
Une seconde recherche est ensuite effectuée avec chacun des vecteurs documents appartenant aux clusters dont les vecteurs centraux sont suffisamment similaires par rapport à la requête.

Comparaison d'une requête à des vecteurs centraux

Cependant, nous verrons que la manière avec laquelle nous confrontons les requêtes et les documents est totalement différente du procédé suivi dans Smart. D'autre part, les procédures documentaires et stratégiques ne sont pas sélectionnées en même temps mais au cours de l'amplification de la portée des requêtes, donc en fonction des réponses apportées par la recherche documentaire.

3. Les sorties désirées

Elles apparaissent sur l'état de sortie suivant. Les livres doivent être rangés par ordre décroissant de pertinence par rapport à la requête présentée.

Identification de l'utilisateur — Date JJ/MM/AA

Requête écrite en langage naturel

Identification des livres proposés (titre,auteur...)	Proximité par rapport à la requête	Proximité par rapport à chacun des objectifs de la requête			
		thème	$objectif_2$	...	$objectif_n$

4. Proximité d'un objectif par rapport à un chemin

A chaque feuille d'une procédure correspond un chemin allant de cette feuille (exclue) à la racine de la procédure (inclue).
Dans une procédure documentaire, la référence d'un livre est située sur une feuille. Pour calculer la pertinence d'un livre par rapport à un objectif d'une requête, il faut donc établir une règle permettant de calculer la plausibilité de proximité d'un chemin par rapport à un objectif.

Soit C un chemin = $obj_1/obj_2/.../obj_n$ constitué de n objectifs reliés d'une façon indépendante à un objectif obj_0 appartenant à une requête :

$$plprox(c,obj_0) = plprox(obj_1,obj_0)//plprox(obj_2,obj_0)//...// plprox(obj_n,obj_0)$$

Chacun des objectifs du chemin est plus ou moins proche de obj_0 et contribue à renforcer la plausibilité de proximité cherchée. La loi de composition interne "parallèle" (//) est donc utilisée.

Exemple$_1$

Soient les liens sémantiques :

(comptabilité, gestion) pd 0.7

(facturation,comptabilité) ass 0.8

plprox(gestion/facturation/édition,comptabilité) =

plprox(gestion,comptabilité)//plprox(facturation,comptabilité)//

plprox(édition,comptabilité) =

(0.9S07)//(0.8S0.8)//0.5 = 0.83

Exemple$_2$

Soient les liens sémantiques :

(automobile,voiture) sd 0.95

(volant,automobile) pd 0.8

plprox(voiture/volant,automobile) =

plprox(voiture,automobile)//plprox(volant,automobile) = 0.87//0.75 = 0.9

5. PROXIMITÉ DE DEUX CHEMINS

5.1. Définition d'un chemin idéal

Un chemin C est idéal pour une requête RQ si toutes les questions composant C sont représentées par tous les objectifs de RQ.

Exemple :

Requête : "Auriez-vous des documents se rapportant à l'attribution de crédits par les banques?"

Traduction :

crédit < attribution
crédit < banque/lieu

Exemple de chemin idéal : banque/crédit/attribution

Mesurer la pertinence d'un livre par rapport à une requête revient donc à déterminer la plausibilité de proximité du chemin correspondant au livre par rapport au chemin idéal de la requête présentée.

5.2. Plausibilité d'atténuation

Si $plprox(a_1,A) = t < 1$,

$plprox(a_1,A < plprox(a_1/B,A/B) < plprox(a_1/B/C,A/B/C)$ ...

Si on doit déterminer la plausibilité de proximité d'un chemin C d'une procédure par rapport à une requête RQ, plus le nombre d'objectifs de RQ est grand, moins la répercussion, sur le résultat cherché, d'une différence entre un objectif de C et un objectif de RQ est grande, ou, ce qui revient au même, plus la gravité de cette différence doit être atténuée lors du calcul de la proximité des deux chemins.

Une "plausibilité d'atténuation ", notée pl-atténuation, est donc définie en fonction de la longueur d'un chemin idéal :

$$\text{pl-atténuation}(N) = 1 - \frac{0.5}{N}$$

N = nombre d'objectifs de la requête

Exemple$_1$:

Si $plprox(a_1,A) = 0.7$ et si $plprox(b_1,B) = 0.8$ et si A/B est le chemin idéal correspondant à une requête, alors

pl-atténuation(2) = 0.75

$plprox(a_1/b_1,A/B) = plprox(a_1/B,A/B)$ S $plprox(A/b_1,A/B)$ =

(0.7//0.75) S (0.8//0.75) = 0.82

Diminuer la gravité d'une différence équivaut à renforcer la plausibilité de proximité cherchée.

Exemple$_2$:

Si $plprox(a_1/A) = 0.7$

alors $plprox(a_1,A) = 0.7//0.5 = 0.7$

$plprox(a_1/B,A/B) = 0.7//0.75 = 0.87$

$plprox(a_1/B/C,A/B/C) = 0.7//0.83 = 0.92$

5.3. Détermination de la plausibilité de proximité de deux chemins

Si C est un chemin appartenant à une procédure stratégique ou documentaire et si CRQ = $obj_1/obj_2/.../obj_n$ est le chemin idéal d'une requête, alors

$$plprox(C,CRQ) = plprox(C,obj_1) \;//\; pl\text{-}atténuation(n)\ S\ plprox(C,obj_2) \;//\; pl\text{-}atténuation(n)\ S\ \ldots\ \ldots\ plprox(C,obj_n) \;//\; pl\text{-}atténuation(n)$$

Exemple :

requête : "Auriez-vous des livres se rapportant à l'automatisation d'une comptabilité?"

Traduction : comptabilité —— automatisation

Chemin idéal : comptabilité/automatisation ⇒ pl-atténuation(2)=0.75

Thésaurus : (comptabilité,gestion) pd 0.8
(cahier-des-charges,automatisation) ass 0.7
(corig,automatisation) moyen 0.9

Un chemin C d'une procédure : C = gestion/cahier-des-charges/corig

plprox(C,comptabilité/automatisation) =
plprox(C,comptabilité) // 0.75 S plprox(C,automatisation) // 0.75 =
[((0.8S0.9) //0.5//0.5)//0.75] S [(0.5//(0.7S0.8)//(0.9S0.9))//0.75]
= 0.88

6. Conditions d'arrêt d'une recherche documentaire

Une recherche documentaire doit être arrêtée quand le système a réussi à satisfaire la requête RQ qui a déclenché cette recherche. Chaque livre considéré pertinent contribue à satisfaire RQ et, par conséquent, provoque le renforcement des plausibilités de proximité de tous les livres proposés et réunis par rapport à RQ et à ses objectifs. A chaque proposition d'un livre L, les plausibilités de proximité par rapport à RQ et à ses objectifs sont donc composées en parallèle avec les plausibilités de proximité des livres déjà proposés par rapport à RQ et à ses objectifs. Lorsque la plausibilité de proximité résultante, notée PLF1, des livres proposés par rapport à RQ, dépasse un $seuil_1$ paramétré, les objectifs de RQ sont considérés atteints.

Cependant, cette condition n'est pas suffisante pour arrêter la recherche : pour la requête précédente, par exemple, ("Auriez-vous des livres se rapportant à l'automatisation d'une comptabilité"?),il est préférable d'avoir un seul livre décrivant l'automatisation d'une comptabilité plutôt que deux documents, l'un décrivant, d'une façon générale, comment est réalisée une automatisation et l'autre ne se rapportant qu'à la description de la comptabilité.

Une nouvelle plausibilité, notée PLF2, a donc été créée pour tenir compte du fait que plus le nombre d'objectifs de RQ traités dans un livre est élevé, plus ce livre est pertinent.

A chaque livre, la plausibilité PLF2 est renforcée.

$$PLF2 := PLF2 \;//\; 0.5 + \frac{0.5}{N} * n$$

N = nombre d'objectifs dans la requête

n = nombre d'objectifs de la requête traités dans le livre proposé

Une recherche est arrêtée si PLF1 > $seuil_1$ et si PLF2 > $seuil_2$ ($seuil_1$ et $seuil_2$ sont deux seuils paramétrés).

Exemple :

requête RQ : "Auriez-vous des livres sur la simulation de la rage?"

tradution : Rage — simulation ⇒ pl.atténuation(2) = 0.75

thésaurus : (rage,épidémie)sd 0.9 ⇒

plprox(rage,épidémie) = 0.8 S 0.9 = 0.75

(épidémie,épidémiologie)ass 0.95 ⇒

plprox(épidémie,épidémiologie) = 0.8 S 0.95 = 0.78

(modèle,simulation)moyen 0.8 ⇒

plprox(modèle,simulation) = 0.8 S 0.9 = 0.75

tests de la satisfaction de la requête au cours de la recherche documentaire:

Description des livres proposés (ordre chronologique)	Plausibilités de proximité des livres proposés			PLF2
	Par rapport à chacun des objectifs de RQ		Par rapport à RQ (PLF1)	
	rage	simulation		
avant le début de la recherche	0.5	0.5	0.5	0.5
livre d'épidémiologie	0.5//0.66 =0.66	0.5//0.5 =0.5	0.5//(0.66//0.75 S 0.5//0.75)=0.69	0.5//0.75 = 0.75
Description des modèles mathématiques	0.66//0.5 = 0.66	0.5//0.75 = 0.75	0.69//(0.5//0.75 S 0.75//0.75) = 0.84	0.75//0.7 = 0.9
Réalisation d'un modèle décrivant l'évolution de la rage vulpine	0.66//1 = 1	0.75//0.75 = 0.9	0.84//(1//0.75 S 0.75//0.75)=0.98	0.9//1 = 1

Si $seuil_1$ = $seuil_2$ = 0.96, la requête est satisfaite et la recherche documentaire est donc arrêtée.

Les conditions d'arrêt définies à la page 117 ne sont pas suffisantes pour prouver la fin de la recherche : en cas d'insuccès de la recherche documentaire, l'amplification de la requête est arrêtée au delà d'un seuil d'amplification paramétré, puis la recherche est interrompue ; si aucun livre ne répond valablement à la requête, un message est imprimé. Sinon la liste des livres qui répondent partiellement à la requête est éditée

7. STRATÉGIE DE LA RECHERCHE DOCUMENTAIRE

7.1. Principe de la stratégie de la recherche documentaire

Rappelons que la recherche documentaire consiste à répondre aux questions contenues dans les procédures documentaires et stratégiques concernées par la requête présentée.

Une fonction d'estimation permet de diriger progressivement la recherche vers les objectifs et les documents qui paraissent être les plus intéressants. (cf. chapitre "ordre de priorité des questions", page 125).

Une réponse à une question Q est affirmative si l'objectif Q a un rapport suffisamment étroit avec la requête, ou, ce qui revient au même, si l'objectif Q fait partie de la liste de tous les objectifs obtenue après l'amplification de la portée de la requête.

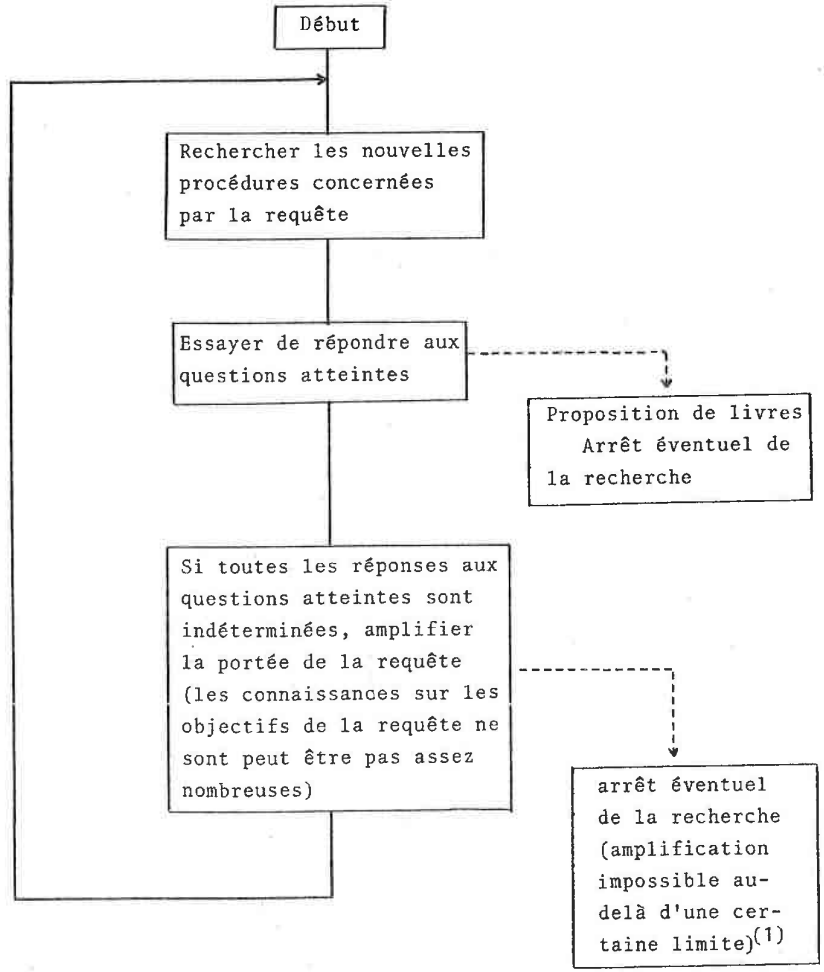
Si le test est positif, le système considère alors les objectifs successeurs de Q. Lorsqu'un livre est atteint, les conditions de sa proposition au lecteur sont remplies.

Par ce principe de recherche, il est évident que plus la portée d'une requête est grande, plus l'exploration a des chances d'aboutir aux feuilles des procédures qui constituent les buts à atteindre. Cependant, l'amplification de la portée d'une requête est un traitement relativement long, donc coûteux, car il nécessite en particulier plusieurs explorations du thésaurus.
Mais, pour satisfaire une requête, il n'est pas toujours indispensable d'associer des objectifs à ceux qui traduisent cette requête.

Exemple :
"J'aimerais avoir un livre contenant des exercices d'initiation à la programmation" est une requête qui peut être directement satisfaite par le livre numéro 4 dans la procédure documentaire "Programmation" (page 180).

Par conséquent, la portée d'une requête ne sera amplifiée qu'en fonction des besoins du module "recherche documentaire".

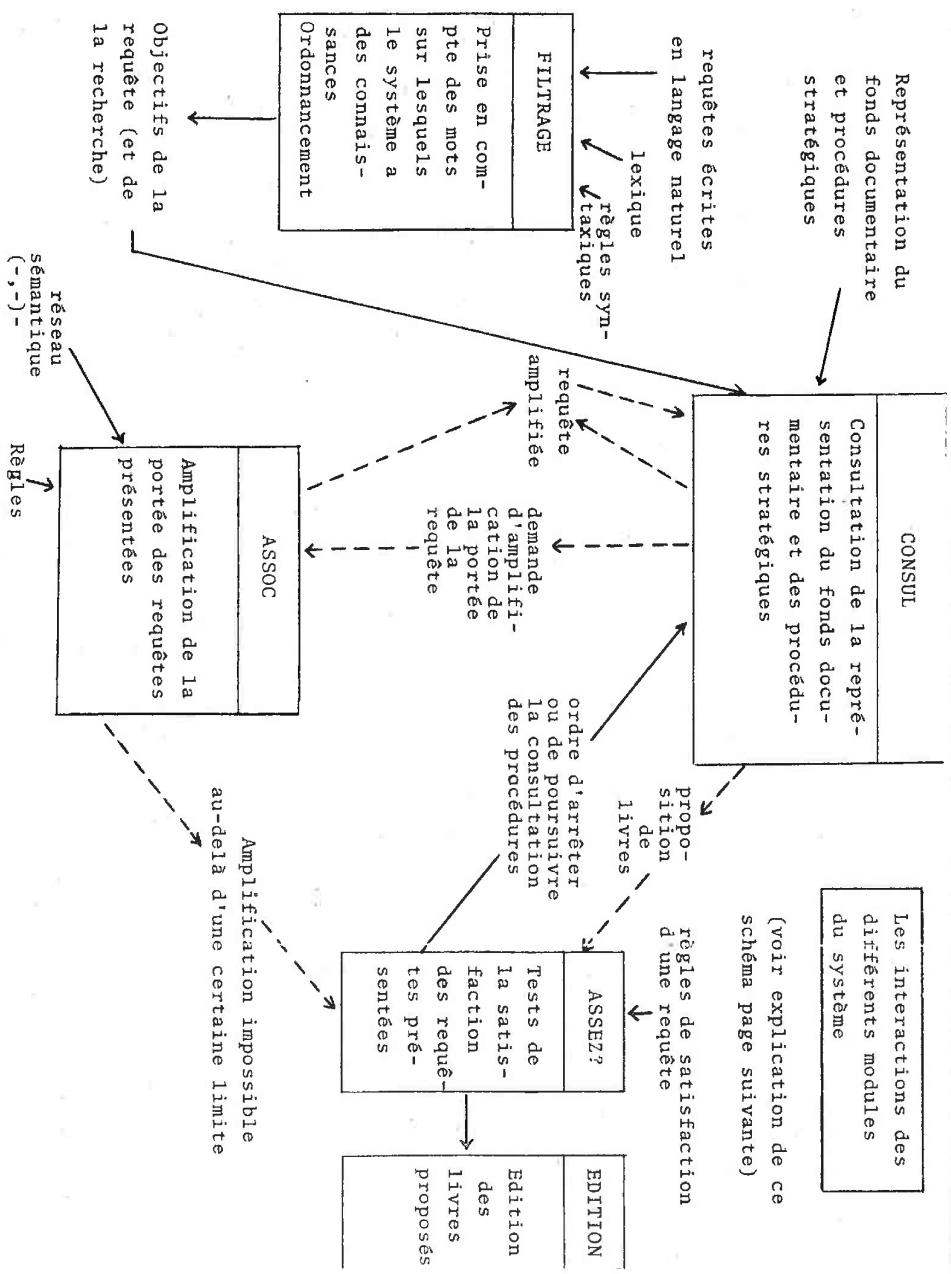
Une recherche documentaire sera donc effectuée en plusieurs étapes :



(1) Le seuil d'amplification maximale, ou seuil de proximité, a été défini page 75 .

Architecture du système :

Compte tenu de la stratégie de la recherche, les interactions entre les différents modules du système sont donc les suivantes :



Interprétation du schéma sur les interactions des différents modules du système :

Les flèches apparaissant en pointillés sur le schéma indiquent que le message n'est envoyé qu'éventuellement, si certaines conditions sont remplies.

On peut distinguer cinq modules :

- FILTRAGE :
  - entrées : - une requête écrite en langage naturel
    - le lexique et les règles syntaxiques
  - fonction : Reconnaître et ordonner les objectifs de la requête
  - sorties : Une liste des objectifs ordonnés de la requête

- CONSUL :
  - entrées : - liste des objectifs ordonnés
    - procédures documentaires et stratégiques
    - état d'avancement de l'exploration dans les procédures
  - fonction : - recherche documentaire
    - enrichir la requête en déterminant l'esprit dans lequel la recherche documentaire doit être effectuée
  - sorties : - éventuellement des livres proposés
    - la requête éventuellement amplifiée
    - les points des procédures atteints par l'exploration

- ASSOC :
  - entrées : - les objectifs de la requête
    - le thésaurus
    - les règles d'association d'un objectif à un autre
  - fonction : Enrichir la requête grâce au réseau des approximations
  - sorties : Une requête dont la portée est amplifiée

- ASSEZ?
  - entrées : - Les adresses et les mesures de la pertinence des livres à proposer
    - Les règles de satisfaction d'une requête
  - fonction : - Déterminer si la requête est satisfaite
  - sorties : - Si la requête est satisfaite
    . ordonner l'arrêt des recherches
    . demander l'édition des livres proposés
    - Sinon : ordonner la poursuite des recherches

- EDITION :
  - entrées : - Référence et pertinence des livres à proposer
  - fonction : - Editer les résultats de la recherche
  - sorties : - Etat des réponses du système à la requête présentée par l'utilisateur

7.2. Ordre de priorité des questions

Appelons chemin exploré, un chemin constitué de questions pour lesquelles le système a répondu affirmativement.
Après un temps t de recherche, à chaque nouvelle question que doit se poser le système correspond un chemin exploré ainsi que les plausibilités de proximité de ce chemin exploré par rapport à chacun des objectifs de la requête.
Il est plus logique et rationnel d'essayer de prolonger les explorations des chemins qui, au temps t, sont les plus prometteuses ou les mieux placées, c'est-à-dire celle qui ont connu le moins d'échecs avant t.
A chaque nouvelle question Q est donc affectée une plausibilité de priorité calculée d'après la proximité du chemin exploré correspondant à Q, par rapport à la requête.
Tous les chemins qui constituent une procédure sont, a priori, aussi prometteurs l'un que l'autre. Avant de débuter leur exploration, leur plausibilité de priorité est donc égale à 1.
Au fur et à mesure de leur exploration, les plausibilités de priorité des questions atteintes tendent à décroître en fonction des échecs de l'exploration. La réponse à une question Q représentée par un objectif peut en effet être un échec dont la gravité est mesurée par l'écart ou la distance sémantique de Q par rapport à l'ensemble des objectifs qui traduisent la requête.

La plausibilité de priorité de Q peut encore être éventuellement affaiblie par une "plausibilité de dépréciation" qui décroît chaque fois qu'une nouvelle tentative pour répondre à Q se solde par une indétermination.

La plausibilité de dépréciation d'une nouvelle question venant d'être atteinte est initialisée à 1. Après chaque tentative infructueuse pour répondre à Q, plausibilité de dépréciation (Q) := plausibilité de dépréciation (Q) S P1 (P1 est un paramètre) ; et plausibilité de priorité (Q) := plausibilité de proximité par rapport à la requête du chemin menant à Q S plausibilité de dépréciation (Q).

7.3. Importance des différentes questions

Plus la plausibilité d'importance d'une question tend vers 0.5, moins cette question est importante.
Si la réponse à une question Q est indéterminée et si la plausibilité d'importance de Q est inférieure à un seuil (noté seuil plimp), le système passe aux questions qui succèdent à Q dans la procédure.

Si un documentaliste ne trouve pas de livres très pertinents, plus la durée de sa recherche est longue, moins il est exigeant sur les conditions à remplir pour proposer les documents les plus pertinents. Le seuil "seuilplimp" sera donc renforcé progressivement au cours de la recherche documentaire :

Nous avons vu (page 121) que la recherche documentaire s'effectue en plusieurs étapes, la fin de chacune d'elles correspondant à une demande d'amplification nécessaire de la requête.

Après chaque étape de recherche :

$$\text{seuilplimp} := \text{seuilplimp} \,//\, p_2$$

p_2 est un paramètre.

Dans le cas d'une insatisfaction prolongée de la requête, le système ne s'attardera donc que sur les questions les plus importantes.

Exemple :

Soit la requête : "j'aimerais savoir comment on programme".
Cette requête a volontairement été choisie pour son imprécision, ce qui empêche le système d'atteindre rapidement des livres pertinents dans la procédure documentaire "programmation" (page 96).
Le lien sémantique (savoir,connaître) vd 0.9 autorise l'exploration de la procédure stratégique "connaître" (page 87).
Dans cette procédure, le système ne peut pas apporter des réponses aux questions juger?, étudier?, réaliser?, compte tenu des connaissances qu'il possède sur la requête.

Initialement, seuil p1 imp := 0.5.
Soit p2 := 0.6.
La requête se prolongeant sans succès, seuil p1 imp croît chaque fois que l'exploration est bloquée, c'est-à-dire dans le cas où il y a indétermination pour toutes les questions atteintes.
Supposons que cette situation se produise 6 fois au cours de la recherche. Seuil p1 imp prend successivement les valeurs suivantes :

0.5//0.6 = 0.6
0.6//0.6 = 0.69
0.69//0.6 = 0.77
0.77//0.6 = 0.83
0.83//0.6 = 0.88
0.88//0.6 = 0.92

Lors du 7ème essai tenté par le système pour poursuivre l'exploration, la question F1 (dans la procédure "connaître") est jugée pas assez importante (0.9 < 0.92) . Le système passe à son successeur : étudier est alors considéré comme un des objectifs de la requête.

7.4. Etude de la stratégie sur un exemple

requête : J'aimerais savoir comment on rédige un dossier de programmation

traduction : adv V N ⇒ N — V — adv (V2)

$V_1\ V_2 \Rightarrow V_2 — V_1$

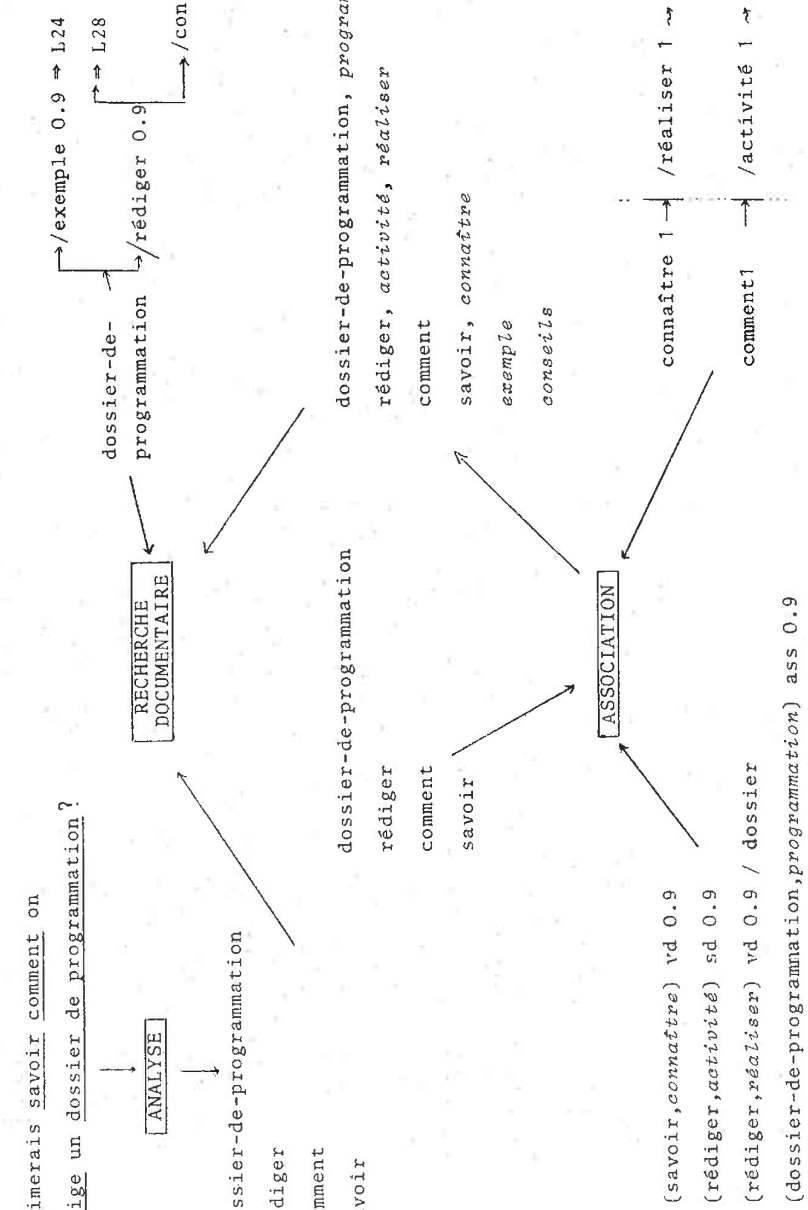
dossier-de-programmation — rédaction — comment — savoir

thésaurus :

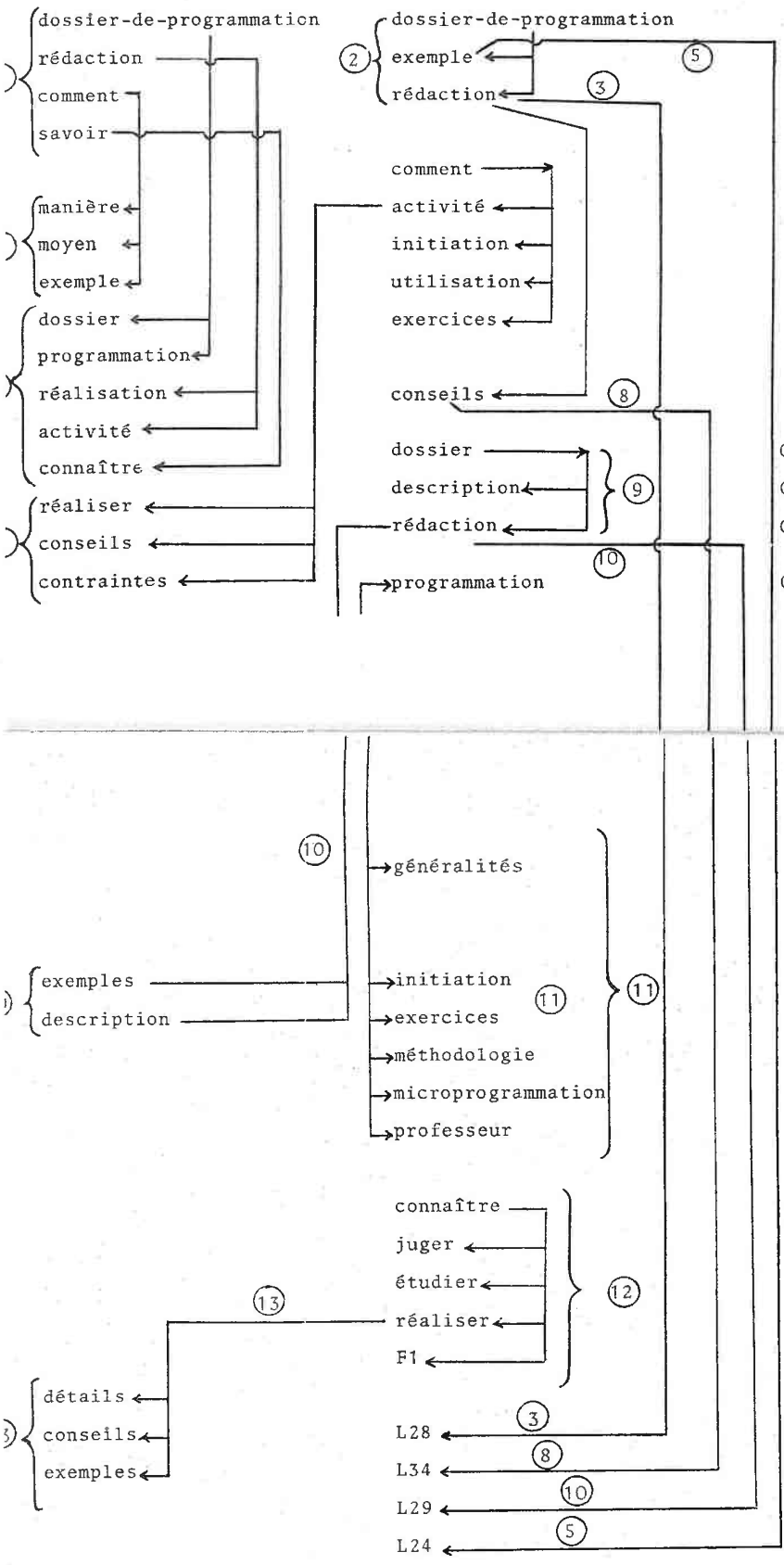
(savoir,connaître)vd 0.95
(rédaction,activité)sd 0.9
(programmation,activité)sd 0.9
(méthode,activité)moyen 0.95
(réaliser,fabriquer)vd 0.8 / objet
(rédaction,réalisation)vdi 0.9
(dossier-de-programmation,programmation)ass 0.95
(dossier-de-programmation,dossier)sd 0.8
(dossier,objet)sd 0.9
(étude-de-cas,exemple)sd 0.9
(critique,avis)sd 0.9
(évaluation,avis)sd 0.9
(date,précis)ass 0.95
(lieu,précis) ass 0.95

La recherche documentaire sera tout d'abord schématisée. Puis son déroulement sera décrit d'après les "traces" et les résultats qui ont été obtenus par la programmation.

DESCRIPTION "STATIQUE" DE LA RECHERCHE DOCUMENTAIRE



REQUETE — EXPLORATION DES PROCEDURES



plausibilité de priorité	dossier de programmation	rédaction	comment	savoir
1	1	0.5	0.5	0.5
1	1	0.5	0.5	0.5
1	1	0.5	0.5	0.5
1	0.5	0.5	1	0.5
1	0.5	0.5	1	0.5
1	0.5	0.5	1	0.5
1	0.5	0.5	1	0.5
1	0.5	0.5	1	0.5
1	1	1	0.5	0.5
0.94 (0.7//0.87)	0.7	0.5	0.5	0.5
0.94 (0.7//0.87)	0.7	0.5	0.5	0.5
0.94 (0.7//0.87)	0.7	0.5	0.5	0.5
0.96 (0.78//0.87)	0.78	0.5	0.5	0.5

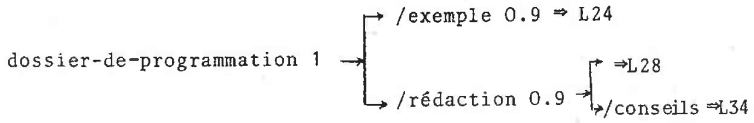
plausibilité de priorité	dossier de programmation	rédaction	comment	savoir
0.96	0.78	0.5	0.5	0.5
0.96	0.78	0.5	0.5	0.5
0.96	0.78	0.5	0.5	0.5
0.96	0.78	0.5	0.5	0.5
0.96	0.78	0.5	0.5	0.5
0.99 (0.95//0.87)	0.5	0.5	0.5	0.95
0.99	0.5	0.5	0.5	0.95
0.99	0.5	0.5	0.5	0.95
0.99	0.5	0.5	0.5	0.95
0.99	0.5	0.5	0.5	0.95
0.8	1	1	0.5	0.5
0.89	1	1	0.9	0.5
0.79	0.96	1	0.5	0.5
0.78	1	0.5	0.75	0.5

Description du déroulement de la recherche documentaire

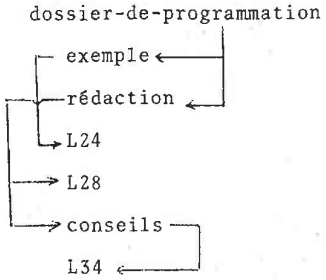
Le schéma précédent fait apparaître clairement le principe de la recherche : le système documentaire confronte la requête aux procédures documentaires et stratégiques concernées par la requête.

La première colonne contient les objectifs qui traduisent initialement la requête, puis les objectifs qui leur sont associés au cours de l'amplification de la portée de la requête.
La deuxième colonne indique linéairement les trajets suivis dans les procédures documentaires et stratégiques, lors de la recherche des documents pertinents.

Exemple :
Soit la procédure documentaire "dossier-de-programmation".



Si le système explore cette procédure et s'il atteint les livres L24, L28, L34, les chemins explorés seront représentés dans la deuxième colonne comme suit :



Le système
est chargé de déterminer les réponses des questions atteintes (2ème colonne) d'après les objectifs de la requête (1ère colonne).

A chaque question Q atteinte correspond une plausibilité de priorité (3ème colonne) et quatre plausibilités de proximité du chemin exploré correspondant à Q par rapport à chacun des quatre objectifs qui traduisent la requête.

Au bas de la page, L28, L34, L29, L24 sont les références des livres considérés pertinents par le système. La proximité de chacun d'eux par rapport à la requête et par rapport à ses objectifs est également indiquée.

Par souci de clarté, les relations et les plausibilités de proximité des objectifs associés par rapport aux objectifs qui traduisent la requête (1ère colonne) ne figurent pas sur le schéma. Les questions devraient constamment être rangées par plausibilités de priorité décroissantes.
Les plausibilités de dépréciation et d'importance ne figurent également pas sur le schéma.

Les nombres entourés indiquent l'ordre chronologique des différentes étapes de la recherche :

① : la recherche ne débute qu'avec les quatre objectifs qui traduisent la requête

② : "dossier-de-programmation" est une procédure documentaire concernée directement par la requête. Le système prolonge donc l'exploration de cette procédure et atteint "exemple" et "rédaction" qui sont les deux questions qui succèdent à la question "dossier-de-programmation" (la procédure documentaire "dossier-de-programmation"figure à la page 101).

③ : la réponse à la question "rédaction" est affirmative puisque "rédaction" est un objectif de la requête. Les successeurs de cette question sont la question "conseils" et le livre L28 qui est donc recommandé au lecteur.
L28 est effectivement pertinent : il s'agit d'un livre intitulé "l'analyse de conception" (MONGORY) qui décrit ce que doit contenir un"dossier de programmation".
Dans l'état actuel des connaissances sur les objectifs de la requête, le système ne peut pas encore savoir s'il est intéressant de montrer des exemples de dossiers de programmation au lecteur.

④ : "comment" est une procédure stratégique qui a un rapport évident avec la requête. Les successeurs de "comment" permettent d'enrichir la requête par les objectifs "manière", "moyen", "exemple" et d'atteindre les questions "activité", "initiation", "utilisation" et "exercices". Il n'est pas encore possible de répondre à ces quatre dernières questions.

⑤ : grâce à la procédure stratégique "comment", l'objectif "exemple" est venu enrichir la liste des objectifs de la requête.
Dans la procédure "dossier-de-programmation", la réponse à la question "exemple" est donc affirmative et le livre L24 est

atteint. Ce livre intitulé "traitements élémentaires en informatique de gestion" (DUSSACQ) contient un exemple de dossier de programmation.

(6) : toutes les réponses aux questions actuelles étant indéterminées, la seule solution pour essayer de prolonger les explorations est d'amplifier la portée de la requête.

(7) : le système sait, grâce aux objectifs associés que la requête a un rapport avec une activité.
Ce renseignement permet donc de répondre affirmativement à la question "activité" (procédure "comment").
Si le lecteur veut savoir comment on réalise une activité, c'est probablement en vue d'être, à son tour, capable de la réaliser lui-même. Il est donc intéressant de lui fournir, en particulier, des conseils. L'objectif "conseils" ainsi que les objectifs "réaliser" et "contraintes" sont donc associés aux objectifs de la requête.

(8) : à la question "conseils", le système répond par conséquent "d'accord" et le livre L34 est atteint.
L34 est un article écrit par G.FONTAINE dans lequel sont décrits certains pièges à éviter, en particulier lors de la rédaction d'un dossier de programmation.
Supposons que la requête ne soit pas encore satisfaite. La recherche documentaire se poursuit.

(9) : dossier est une procédure ayant un rapport avec la requête. Elle est donc explorée et les deux questions "description" et "rédaction" sont posées au système.

(10) : "rédaction" est un objectif de la requête. Le système répond donc "oui" à la question "rédaction" qui figure dans la procédure "dossier".
La requête est enrichie et le livre L29 est proposé au lecteur (L29 : Présentation d'un rapport de stage (Mongory)).
Supposons que le système considère qu'il a satisfait la requête.

La recherche est arrêtée.
D'autres procédures telles que "programmation" ((11)) et "connaître" ((12)) auraient cependant pu être explorées, ce qui aurait peut être permis la proposition d'autres livres et un nouvel enrichissement de la requête ((13)).

8. LES PROCÉDURES STRUCTURALES

8.1. Définition

Certaines questions ne sont pas des objectifs du lexique (exemple : "langage particulier?" dans la procédure documentaire programmation (page 96) .

Nous verrons que leur réponse est déterminée d'après la structure de la requête présentée ou, plus précisément, d'après la nature des relations qui relient les objectifs qui représentent la requête. Les questions concernant la structure d'une requête peuvent être très diverses. Nous avons seulement retenu les types de question qui nous paraissent être les plus intéressants.
A chaque type de question correspond dans le programme informatique une procédure structurale dont le rôle est de déterminer la réponse à la question qui provoque son appel.

8.2. Liste des procédures structurales

- PARTICULIER(OBJ) ;

  Le lecteur s'intéresse-t-il à un obj particulier ?
  exemple : particulier (langage-de-programmation)
  (pour faciliter la compréhension, nous avons employé dans les procédures documentaires et stratégiques décrites, la forme plus naturelle "langage(de-programmation) particulier")

- GENERALISATION (OBJ)

  Le lecteur est-il intéressé par une généralisation de obj ?
  Exemple : généralisation (Cobol)

- UTILE (OBJ) ;

  Le lecteur désire-t-il savoir ce que permet de faire obj ?
  Exemple : utile(éditeur-de-lien)

- REALISATION(OBJ) ;

  Le lecteur veut-il savoir comment est réalisé obj ?
  Exemple : réalisation(automatisation)

- VOISIN (OBJ) ;

  Le lecteur désire-t-il étudier des objectifs voisins de obj ?
  Exemple : voisin(temps-réel)

- CONSTITUE (OBJ) ;

  Le lecteur veut-il étudier l'objectif dont obj est une partie incluse ?
  Exemple : constitué (bloc commande)

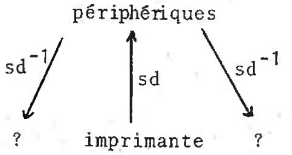
- CONSTITUANT (OBJ) ;

  Le lecteur s'intéresse-t-il aux constituants de obj ?
  Exemple : constituant (unité-centrale)

- AUTRE-PARTICULIER (OBJ) ;

Le lecteur s'intéresse-t-il à d'autres obj particuliers ?

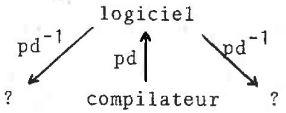
Exemple : autre-particulier (imprimante)



- AUTRE-CONSTITUANT (OBJ) ;

Le lecteur s'intéresse-til à d'autres constituants que obj ?

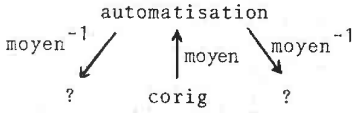
Exemple : Autre-constituant (compilateur)



- AUTRE-MOYEN (OBJ) ;

Le lecteur s'intéresse-t-il à d'autres moyens que obj ?

Exemple : autre-moyen (corig)



8.3. Exemple de détermination de réponses à des questions qui appellent des procédures structurales

requête : "Je voudrais concevoir des statistiques sur des ventes : auriez-vous des livres se rapportant à ce sujet? "

traduction :

règles : V N $\Rightarrow$ N — V

traitement sur action $\Rightarrow$ action — traitement

ventes — statistiques — conception

thésaurus :

(analyse,conception) moyen 0.9 / gestion

(programmation,réalisation) moyen 0.9 / gestion

(statistiques,traitement) sd 0.95

(ventes,application) sd 0.9 / gestion

(ventes,gestion) ass 0.7

REQUETE — EXPLORATION DES PROCEDURES

objectifs de la requête			questions atteintes	plausibilité de priorité	ventes	statistiques	conception
ventes	vd	1					
statistiques	vd	1					
conception	vd	1					
gestion	ass	0.64					
application	sd^{-1}	0.83					
traitement	sd^{-1}	0.91					
analyse	moyen	0.83					
			analyse	0.96	0.5	0.5	0.83
			généralités	0.96	0.5	0.5	0.83
			initiation	0.96	0.5	0.5	0.83
			exercices	0.96	0.5	0.5	0.83
			analyse-fonctionnelle	0.96	0.5	0.5	0.83
			analyse-organique	0.96	0.5	0.5	0.83
			méthodologie	0.96	0.5	0.5	0.83
			conseils	0.96	0.5	0.5	0.83
			professeur	0.96	0.5	0.5	0.83
			chef-de-projet	0.96	0.5	0.5	0.83

			détail	0.96	0.5	0.5	0.83
			traitement	0.96	0.5	0.5	0.83
			application	0.96	0.5	0.5	0.83
			organisation	0.94	0.5	0.91	0.83
			traitement particulier	0.94	0.5	0.91	0.83
			application particulière	0.92	0.83	0.5	0.83
			saisie	0.94	0.5	0.91	0.83
			contrôle	0.94	0.5	0.91	0.83
			mise-à-jour	0.94	0.5	0.91	0.83
			édition	0.94	0.5	0.91	0.83
			statistiques	0.94	0.5	0.91	0.83
			production	0.92	0.83	0.5	0.83
			stock	0.92	0.83	0.5	0.83
			facturation	0.92	0.83	0.5	0.83
			ventes	0.92	0.83	0.5	0.83
			ventes	0.96	0.5	1	0.83
			L14	0.78	0.83	0.5	0.83
			L22	0.96	1	1	0.83

Commentaires :

(1) Il est nécessaire d'amplifier la portée de la requête pour atteindre une procédure qui s'y rapporte.

(2) La procédure "analyse" est explorée. Parmi les questions atteintes figurent "traitement" et "application".

(3) Les réponses à ces deux questions sont affirmatives puisque "traitement" et "application" sont deux objectifs de la requête.
Le système doit ensuite se poser les questions suivantes: le lecteur s'intéresse-til à un traitement particulier et à une application particulière ?

(4) Les réponses à ces deux questions sont encore affirmatives car, dans la requête, les relations associées à "application"et à "traitement" sont égales à sd^{-1}. Elles indiquent que le lecteur veut étudier un traitement particulier (statistiques) sur une application particulière (ventes).

(5) La recherche aboutit à la proposition de deux livres dont l'un (L22) est très pertinent puisque les objectifs "ventes" et "statistiques" sont parfaitement atteints et que "conception" l'est à 83%.

9. La procédure Spécialité

Tout comme les procédures structurales, SPECIALITE est une procédure qui appartient au programme informatique.
Quand elle est appelée, son rôle est de déterminer si la requête présentée se rapporte au domaine de spécialité du lecteur.
Cette information est intéressante dans la mesure où elle aiguille la recherche vers des livres correspondant généralement au niveau du lecteur. La réponse à la question "spécialité"est déterminée d'après la proximité du domaine de spécialité du lecteur par rapport aux objectifs de la requête.

10. La procédure MESSAGE(TX); TEXT TX ;

Cette procédure est encore écrite dans le programme informatique. Elle peut également être appelée depuis les procédures documentaires et stratégiques.
Elle permet d'envoyer des messages aux utilisateurs.
Exemple : MESSAGE ('PAS DE LIVRE EN ECONOMIE');

11. Conclusion

Les questions à intégrer dans les procédures documentaires et stratégiques peuvent donc porter :

- sur les objectifs informatiques exprimés dans la requête
- sur les objectifs désignant le cadre du sujet informatique de la requête. Les renseignements contextuels peuvent concerner :

  . les objectifs non informatiques de la langue française servant à exprimer ce que veut exactement le lecteur à propos du sujet informatique de sa requête

  . le lecteur lui-même. La connaissance de son domaine de spécialité ainsi que la nature de sa profession oriente la recherche vers des documents simples ou complexes, détaillés ou superficiels, qui sont hermétiques ou qui vulgarisent le travail des spécialistes...

- sur la structure de la requête.

VI. PROGRAMMATION

VI. PROGRAMMATION

1. INTRODUCTION

Le système est implanté. Nous tentons diverses manipulations expérimentales en vue de tester l'opportunité de tous les mécanismes décrits.

1.1. Choix du langage de programmation

Le programme informatique est écrit en Simula. Ce langage de simulation a été choisi car il nous semble bien adapté aux traitements de textes et de listes:
La classe système SIMSET contient des procédures permettant des traitements de listes relativement aisés.
Des listes sont en particulier utilisées lors de la recherche documentaire au cours de laquelle la liste des questions atteintes dans les procédures stratégiques et documentaires est confrontée à la liste des objectifs qui représentent la requête.

1.2. Les étapes de la programmation

Deux phases successives apparaissent nécessairement :

- la création des connaissances
- la recherche documentaire qui consiste principalement en une exploration des connaissances acquises.

2. LA CRÉATION DES CONNAISSANCES

Rappelons que les connaissances sont principalement le lexique, le thésaurus, les procédures stratégiques et documentaires.

2.1. La création du lexique :

2.2.1. Représentation interne

Le tableau TEXTOBJ contient le libellé des objectifs. Chacun est représenté à l'aide de 17 caractères au maximum.
déclaration : TEXT ARRAY TEXTOBJ(1:MAXLEXIQUE) ;
MAXLEXIQUE est un paramètre.

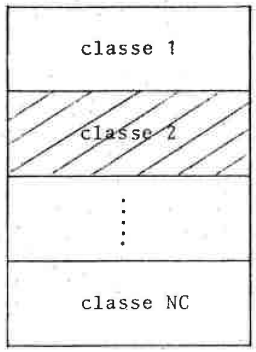
Les adresses de fin des procédures documentaires et stratégiques figurent dans le tableau LEXIQUE. Ce dernier permet donc de savoir quels sont les objectifs qui sont des procédures et où se situent ces procédures.
Les adresses sont rangées dans le tableau LEXIQUE lors de la création des procédures.
déclaration : INTEGER ARRAY LEXIQUE (1:MAXLEXIQUE) ;

2.1.2. Organisation du lexique

Des classes non sémantiques d'objectifs ont été formées, ce qui permet d'accélérer les recherches dans le lexique, en n'explorant que la classe à laquelle appartient l'objectif recherché.

De telles recherches sont effectuées par le système documentaire quand, par exemple, il contrôle qu'un mot appartenant à une requête venant d'être lue, fait partie du vocabulaire sur lequel il a des connaissances et constitue donc un objectif.

Subdivision du lexique en classes :



partie du lexique à explorer pour rechercher un objectif appartenant à la classe 2

Le temps de recherche est donc environ divisé par NC.

2.1.2.1. Détermination de la classe d'un objectif D

- Soit NC le nombre de classes désiré dans le lexique (NC est un paramètre)
- Soit VALOBJ un nombre correspondant à D calculé comme suit :

$$VALOBJ := \sum_{i=1}^{n} i * \text{valeur d'implémentation du ième caractère de D}$$

(n est un paramètre ≤ 17)

Le numéro (CLASOBJ) de la classe D est égal à :

CLASOBJ := VALOBJ -(VALOBJ / NC)* NC + 1 ;

↑ division entière

2.1.2.2. Calcul des adresses de début et de fin des classes

Les adresses de fin (par déduction les adresses de début) sont rangées dans le tableau ADLEXIQUE.

déclaration : INTEGER ARRAY ADLEXIQUE(1:NC) ;

Lors de la création du lexique, la classe de chaque objectif est calculée.
Au terme de la création, on obtient le nombre d'objectifs par classe.

Exemple :

Nombre d'objectifs du lexique par classe

classe 1	2
classe 2	2
classe 3	3
classe 4	2
classe 5	2
classe 6	2
classe 7	3
classe 8	4
classe 9	4
classe 10	4
classe 11	3
classe 12	1
classe 13	4
classe 14	2
classe 15	1

Le lexique est ensuite trié par numéros de classe croissants. Les nombres d'objectifs par classe sont cumulés. Ils représentent alors les adresses de fin des classes dans le lexique.

Adresses de fin des classes dans le lexique

classe 1	2
classe 2	4
classe 3	7
classe 4	9
classe 5	11
classe 6	13
classe 7	16
classe 8	20
classe 9	24
classe 10	28
classe 11	31
classe 12	32
classe 13	36
classe 14	38
classe 15	39

2.2. La création du thésaurus

Organisation du thésaurus

Le format d'un lien sémantique étant fixe, le thésaurus est représenté à l'aide du tableau à 5 colonnes THESAURUS (déclaration : INTEGER ARRAY THESAURUS (1:P,1:5) (P est le paramètre indiquant le nombre maximum de liens sémantiques souhaités dans le thésaurus).

Lors de l'amplification de la portée d'une requête, pour pouvoir associer des objectifs à un objectif obj_i donné, il faut tout d'abord rechercher les liens sémantiques qui contiennent obj_i et grâce auxquels des associations pourront éventuellement être effectuées. Le thésaurus est organisé d'une façon telle que la recherche d'un objectif soit rapide.

Les objectifs à rechercher figurent dans la première ou dans la deuxième colonne du thésaurus.

Tout comme pour le lexique, la première colonne est répartie en NC classes non sémantiques.

L'objectif relié à un objectif de la première colonne peut appartenir à une quelconque des NC classes. En conséquence, à chaque classe dans la première colonne correspond un ensemble d'objectifs dans la deuxième colonne qui doit être lui-même réparti en NC classes.

Exemple : Si NC = 3, les deux premières colonnes du thésaurus seraient organisées de la façon suivante :

classe 1	classe 1
	classe 2
	classe 3
classe 2	classe 1
	classe 2
	classe 3
classe 3	classe 1
	classe 2
	classe 3

partie du thésaurus à explorer pour rechercher un objectif appartenant à la classe 2

2.2.1. Contrôles sur la validité des liens sémantiques

Lors de la lecture du fichier destiné à la création du thésaurus, des contrôles sont effectués :

- sur les objectifs : on teste leur présence dans le lexique. Un objectif valide est remplacé dans le tableau du thésaurus par son adresse dans le lexique. Lorsqu'un domaine de validité est absent dans un lien sémantique, on indique un 0 (zéro) dans la 5ème colonne de THESAURUS.

- sur les relations : leur libellé doit être égal à l'un de ceux qui sont contenus dans le tableau de textes RELATION. Chaque relation valide est remplacée dans THESAURUS par son adresse dans le tableau RELATION.

- sur les plausibilités d'intérêt : leur valeur doit appartenir à l'intervalle [0,1]. Si c'est le cas, la plausibilité est multipliée par 100 avant d'être rangée dans la quatrième colonne

de THESAURUS (le nombre de chiffres décimaux d'une plausibilité d'intérêt est limité à 2. La multiplication par 100 a pour effet la transformation du nombre réel en un nombre entier).

Toute carte erronée est rejetée et imprimée avec le libellé correspondant à l'erreur. Les nombres entiers obtenus dans le tableau THESAURUS ont l'avantage d'être moins encombrants et plus faciles à manipuler que des textes.

2.2.2. Subdivision du thésaurus en classes

but : effectuer des recherches d'objectifs plus rapides
détermination de la classe d'un objectif D : même règle que celle de la page 153.

Calcul des adresses de début et de fin des classes du thésaurus

Les adresses de fin (par déduction les adresses de début) sont rangées dans le tableau ADTHESAURUS.
déclaration : INTEGER ARRAY ADTHESAURUS(1:NC,1:NC) ;

Lors de la création du thésaurus, après la lecture d'un lien sémantique valide (par exemple : (objectif$_i$,objectif$_j$)R,pl(objectif$_k$)

- la classe I de l'objectif$_i$ est calculée puis celle (J) de l'objectif$_j$
- le nombre | I | J | adresse de l'objectif$_j$ dans le lexique |
  est provisoirement enregistré à la place de l'objectif$_j$ dans la deuxième colonne de THESAURUS.
- la case ADTHESAURUS(I,J) est incrémentée de 1.

Au terme de la lecture des liens sémantiques, on obtient le nombre d'objectifs de chacune des NC^2 parties de classe qui figurent dans la deuxième colonne de THESAURUS.

Exemple :

Si NC = 15, exemple de nombres d'objectifs dans chacune des NC^2 parties de classe :

Effectif des classes du thésaurus :

	classe n°	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
classe n°	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	2	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	3	0	0	1	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0	4	0	0	1	0	1	0
	13	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	14	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Le thésaurus est ensuite trié suivant les nombres croissants de la deuxième colonne de THESAURUS.
A la fin du tri, les nombres | I | J | sont ôtés : chaque objectif n'est plus désigné dans le thésaurus que par son adresse dans le lexique.

Les nombres du tableau ADTHESAURUS sont ensuite cumulés : ils représentent alors les adresses de fin des NC parties de chacune des NC classes de la deuxième colonne de THESAURUS.

Exemple déduit du tableau précédent :

Adresses de fin des classes du thésaurus :

	classe n° 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
classe n° 1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	7	7	7	7
5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	7	7	7	7	7	7	7	7	7	7	8	8	8	8	8
7	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	8	8	8	8	8	8	8	8	11	11	11	12	12	12	12
10	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
11	12	12	12	13	13	13	13	13	13	13	13	13	13	13	13
12	13	13	13	13	13	13	13	13	17	17	17	18	18	19	19
13	19	19	19	19	19	19	19	19	19	19	20	20	20	20	20
14	20	20	21	22	22	22	22	22	22	22	22	23	23	23	23
15	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23

2.3. La création des procédures stratégiques

Les procédures sont, dans un premier temps, transcrites sur cartes. Grâce au fichier obtenu, les procédures stratégiques sont ensuite rangées dans le tableau TABSTRAG.

Déclaration : INTEGER ARRAY TABSTRAG(1:MAXPROC,1:4) ;
(MAXPROC est un paramètre indiquant le nombre total d'objectifs contenus dans les procédures stratégiques)

2.3.1. Transcription sur cartes

Les procédures doivent être lues suivant l'ordre breadth-first et transcrites au fur et à mesure sur cartes.
Chaque question est suivie de sa plausibilité d'importance et de son nombre de successeurs.
Une plausibilité d'intérêt et un nombre de successeurs égal à 0 figurent après chaque objectif pouvant être associé à ceux qui expriment la requête.

Exemple : procédure stratégique "comment" (page 91)

Comment 1;7 manière 0.9;0 moyen 0.9;0 exemple 0.9;0
activité 1;3 initiation 1;1 utilisation 1;1 exercices 1;2
réalisation 0.8;0 conseils 0.9;0 contraintes 0.7;0 étude-schématique 0.8;
mode-d'emploi 0.9;0 exsupcours 0.9;0 excorrigés 0.9;0

2.3.2. Contrôles et rangement des procédures stratégiques :

Lors de la lecture des cartes perforées, on teste la validité de chaque objectif en vérifiant sa présence dans le lexique. On contrôle également que chaque plausibilité d'importance ou d'intérêt a une valeur appartenant à l'intervalle [0,1] et que le nombre de successeurs est une valeur numérique positive ou nulle. Toute erreur est signalée et provoque le rejet de la procédure. Si aucune erreur n'est détectée, pour chaque objectif, on indique

- dans la 1ère colonne de TABSTRAG, son adresse dans le lexique
- dans la 2ème colonne, son nombre de successeurs
- dans la 3ème colonne, l'adresse de son premier successeur dans TABSTRAG
- dans la 4ème colonne, sa plausibilité d'importance ou d'intérêt

Exemple : procédure "comment" (en gardant la forme externe des objectifs)

n° de ligne	objectifs	nombres de successeurs	adresse du premier successeur	plausibilité d'importance ou d'intérêt (*100)
1	comment	7	2	100
2	manière	0	0	90
3	moyen	0	0	90
4	exemple	0	0	90
5	activité	3	9	100
6	initiation	1	12	100
7	utilisation	1	13	100
8	exercices	2	14	100
9	réalisation	0	0	80
10	conseils	0	0	90
11	contraintes	0	0	70
12	étude-schématique	0	0	80
13	mode-d'emploi	0	0	90
14	exsupcours	0	0	90
15	excorrigés	0	0	90

Les procédures valides sont rangées dans TABSTRAG qui est donc partitionné. Le nom d'une procédure représente le point d'entrée unique de son exploration. Soit obj un objectif représentant une procédure stratégique : on associe à obj dans le lexique, l'adresse du point d'entrée de obj dans TABSTRAG.

2.3.3. Edition des procédures stratégiques

Les états de sortie édités doivent être rendus compréhensibles. L'édition est effectuée à partir du tableau TABSTRAG.

Exemple : édition de la procédure stratégique "comment"

```
comment 1
        manière 0.9
        moyen 0.9
        exemple 0.9
        /activité 1
                réalisation 0.8
                conseils    0.9
                contraintes 0.7
        /initiation 1
                étude-schématique 0.8
        /utilisation 1
                mode-d'emploi 0.9
        /exercices 1
                exsupcours 0.9
                excorrigés 0.9
```

2.4. La création des procédures documentaires

Les procédures documentaires sont rangées dans le tableau TABFONDS. La transcription sur cartes, les contrôles de validité, l'indication des adresses des procédures documentaires ainsi que l'édition du fonds documentaire s'effectuent suivant le même principe que pour les procédures stratégiques.

Cependant, une feuille d'une procédure documentaire peut être l'adresse d'un livre ou d'un message pouvant être envoyé aux utilisateurs.

Deux autres tableaux sont donc nécessaires :
- l'un contient les données signalétiques des livres
- dans l'autre figurent les libellés des différents messages autorisés.

3. LA RECHERCHE DOCUMENTAIRE

3.1. Les sorties désirées

L'état de sortie à produire est dessiné page 110.

3.2. Les entrées du module "recherche documentaire"

Elles sont constituées, d'une part, par les connaissances possédées par le système et qui viennent d'être décrites, d'autre part, par la requête écrite en langage naturel et suivie de la liste de ses objectifs ordonnés :
Avant de tester l'analyseur syntaxique chargé d'ordonner les objectifs, nous voulons en priorité vérifier la validité de l'analyse sémantique des requêtes ainsi que la qualité des résultats des recherches documentaires. Par conséquent, l'utilisateur doit indexer manuellement sa requête en choisissant dans le lexique, les objectifs qui traduisent le mieux sa signification.

Rappelons qu'il ne doit pas pondérer ou relier les objectifs :
- les relations entre objectifs ne sont pas toujours évidentes et leur indication entraînerait un effort supplémentaire pour traduire les requêtes et acquérir davantage de connaissances sur la façon de s'y prendre.

- afin de faciliter dans un premier temps les traitements documentaires, des opérateurs tels que "et", "ou",

"sauf" ne sont pas non plus introduits dans les connaissances du système. Ils ne doivent donc pas être insérés entre les objectifs qui indexent une requête. L'opérateur "et" relie implicitement chacun des objectifs.

Si une requête comporte des conjonctions et/ou des négations, il convient donc de la modifier ou de l'éclater en plusieurs requêtes affirmatives.
Une requête devient plus "exigeante" si un opérateur "ou" n'est pas considéré.
Par contre, si le système ne prend pas en compte une négation, ses réponses seront fausses puisque c'est l'opposé de la négation qu'il considère.

Exemples de types de requêtes à ne pas présenter :

- Auriez-vous des livres d'informatique de gestion qui ne se rapportent [pas] à la comptabilité ou à la gestion commerciale ?

- Auriez-vous des livres dans lesquels il y aurait des exercices de programmation [sans] leur corrigé ?

- Auriez-vous des livres dans lesquels il y aurait des généralités sur les langages de programmation [sauf] le fortran et le PL/1 ?

Il est prévu de lever prochainement ces restrictions.

3.3. Contrôles sur la validité des requêtes

Afin d'éviter des traitements longs et sans intérêt, le système doit s'assurer tout d'abord qu'il peut a priori traiter chaque requête présentée. Une recherche documentaire est en effet inutile si les objectifs d'une requête ne sont pas tous connus.

Il est donc nécessaire d'effectuer des contrôles sur la validité de chacun des objectifs d'une requête présentée.
Si aucun objectif n'est erroné, la requête est acceptée.
S'il y a au moins un objectif non valide, la requête est rejetée.
Pour chacun des objectifs erronés, un message d'erreur est imprimé.

3.4. La stratégie de la recherche

Au cours de la recherche documentaire, les procédures stratégiques et documentaires sont explorées suivant la stratégie qui a été reproduite telle qu'elle a été décrite lors de sa conception.

4. CONCLUSION

Les résultats qu'obtiendra le système réalisé permettront de faire une étude d'évaluation, en particulier grâce aux deux ratios "Rappel" et "Précision" définis à la page 68 .

VII. CONCLUSION

GENERALE

VII. CONCLUSION GENERALE

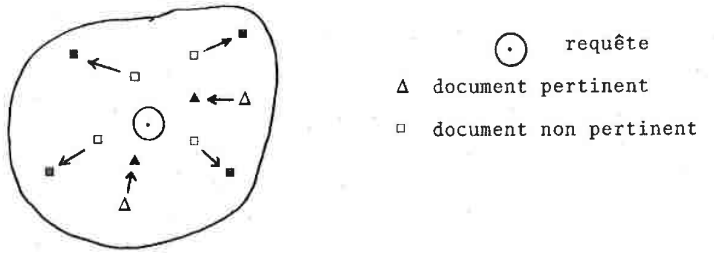
Le système que nous venons de décrire n'est pas parfait :

Il ne faut pas oublier qu'un livre contient généralement plusieurs dizaines de pages et n'est représenté dans notre système que par quelques informations.

Ce minimum d'informations n'est pas susceptible de rendre compte de toute la richesse des documents mais il présente l'avantage de permettre une recherche relativement rapide et aisée des références bibliographiques qui paraissent être les plus pertinentes pour une requête présentée.

Plusieurs traitements documentaires possibles en mode conversationnel ne sont pas réalisés par notre système : citons, par exemple, l'ajustement de la représentation d'un document proposé en fonction de sa pertinence évaluée par le lecteur [28](1)

(1) Si un document est pertinent, sa représentation est "rapprochée" de celle de la requête (et inversement)



Ajustement de la représentation des documents

Soit q la représentation d'une requête et d celle d'un document pertinent.

- un $concept_i$ présent dans q et absent de d est ajouté avec le poids β (paramétré)
- le poids du $concept_i$ présent dans q et d est accru
- le poids d'un $concept_i$ présent dans d et absent de q décroît.

Pour modifier les documents non pertinents, les actions à entreprendre sont inversées.

Cet ajustement permanent de la représentation des documents permet d'améliorer les performances des recherches documentaires ultérieures.

En mode interactif, l'effort de l'utilisateur est fourni lors de la recherche documentaire. Les résultats dépendent alors beaucoup de la façon dont le chercheur accomplit celle-ci.

Notre système peut suggérer et contrôler l'amplification de la portée des requêtes, ce qui pourrait impliquer peut être moins d'efforts de la part des futurs utilisateurs.

Cependant, nous avons vu, lors de la présentation de notre étude (chapitre II) que notre ambition n'a jamais été de construire un système documentaire capable de concurrencer des systèmes tels que Satin , Smart ou Mistral : nous avons en effet voulu éviter la rigueur d'un langage documentaire formel et, au contraire, garder la diversité d'interprétation du Langage Naturel servant à formuler les requêtes et à représenter les documents.

Rappelons que nos objectifs étaient principalement d'étudier l'intégration et l'utilisation des connaissances d'un documentaliste dans un système documentaire. Pour tenter d'atteindre ces objectifs nous avons défini un modèle de représentation et d'exploration des connaissances, du genre de ceux employés en Intelligence Artificielle.

En écrivant les connaissances sur la matière documentaire, nous nous sommes efforcés de ne pas nous limiter à une simple classification des domaines. Nous avons plutôt tenté de représenter les documents en fonction de besoins précis qu'ils pourraient satisfaire (ex : initiation, entraînement, perfectionnement...).

Les connaissances portent aussi sur les lecteurs et sur des termes généraux tels que :exercices, connaître, comment...

A l'aide de procédures stratégiques et d'un réseau des approximations (caractérisées par des plausibilités), nous essayons de découvrir les informations qui sont sous-entendues dans les requêtes.

Grâce à ces outils, les tests effectués lors de la recherche documentaire pour passer d'un domaine à un autre, ne sont pas brutaux et figés puisqu'ils peuvent faire appel à des connaissances contextuelles qui permettent des réactions différentes suivant l'environnement.

De plus, ces outils nous permettent d'évaluer la pertinence d'un document par rapport à une requête et à chacun de ses objectifs.

Tous les mécanismes décrits sont implantés. Pour établir une comparaison valable avec d'autres systèmes, il conviendrait maintenant d'effectuer une expérimentation à plus grande échelle en prenant en compte des requêtes venant d'utilisateurs neutres.

ANNEXES

Pour illustrer notre travail, nous allons analyser quelques résultats de programmation, à l'aide desquels nous désirons montrer le rôle joué par :

- les procédures stratégiques et par les plausibilités d'importance (annexe 1)
- les autres types de plausibilité (annexe 2).

L'analyse va porter sur les traitements de deux requêtes :

- la première : "j'aimerais savoir comment on programme?" est exagérément évasive, ce qui permettra de tester les stratégies d'amplification.
- la deuxième : "je recherche un recueil d'exercices pour apprendre la programmation?" est plus classique et servira de référence.

Les documents pouvant être proposés sont les suivants:

n° d'ordre des documents	Titres	Auteurs
1	Introduction to computer programming	KNUTH
2	La programmation systématique	WIRTH
3	Programming teaching techniques	TURSKI
4	Problems for computer solutions	GRUENBERGER
5	The art of computer science	FORSYTHE
6	Programmation modulaire	MAYNARD
7	Méthodologie	MAROLDT
8	La microprogrammation	MEDARD
9	Problèmes de microprogrammation	PINSON
10	Microprogramming	BOULAYE

- Trois procédures seront utilisées : une,documentaire, pour l'objectif 'programmation', et deux,stratégiques,associées à 'connaître' et 'comment'.

ANNEXE N°1

Pour évaluer le rôle joué par les procédures stratégiques et par les plausibilités d'importance, nous allons observer l'effet que produit la prise en compte progressive de ces deux types d'information.

Les deux requêtes
- j'aimerais savoir comment on programme?
- je recherche un recueil d'exercices pour apprendre la programmation?

vont être traitées quatre fois :

1er cas : Traitements sans procédure stratégique et en neutralisant le mécanisme des plausibilités d'importance.

2ème cas: Pas de procédure stratégique ; les plausibilités d'importance des questions de la procédure documentaire sont plus ou moins élevées en fonction de l'importance accordée aux différentes conditions d'accès aux documents.

3ème cas: Nous ajoutons aux connaissances utilisées dans le deuxième cas, les procédures stratégiques dans lesquelles les plausibilités d'importance fournies sont sans effet sur les résultats.

4ème cas: Les plausibilités d'importance associées aux questions des procédures stratégiques et documentaires sont plus ou moins élevées.

Auparavant, nous allons montrer une partie du lexique puis le thésaurus qui ont aidé le système à effectuer les recherches documentaires.

Objectifs ayant été référencés pour

l'interprétation des requêtes

objectifs	adresse procédures associées	objectifs	adresse procédures associées
organisation		exemple	
initiation		réaliser	
étudier		réalisation	
performance		manière	
professeur		description	
programmation	35	application	
moyen		apprendre	
connaître	69	savoir	
juger		traitement	
généralité		utilisation	
F1		précis	
critique		imprécis	
conception		gestion	
spécialité		méthodologie	
conseils		mode d'emploi	
rapport		analyse	
évaluation		microprogrammatio	
activité		comment	
réalisation		rédaction	
exercice		objet	

THESAURUS :
****

OBJECTIF 1	OBJECTIF 2	RELATION	PLAUSIBILITE	DOMAINE VALIDITE
**********	**********	********	**********	**********
SAVOIR	CONNAITRE	VD	095	
REDACTION	REALISER	VD	095	
REDACTION	ACTIVITE	SD	095	
MOYEN	METHODOLOGIE	SD-1	080	PROGRAMMATION
APPRENDRE	CONNAITRE	MOYEN	095	PROGRAMMATION
EXEMPLE	EXERCICE	ASS	080	

FONDS DOCUMENTAIRE
****

1er cas : les deux requêtes vont être traitées sans procédure stratégique et avec la procédure documentaire ci-contre

```
PROGRAMMATION      0.50
      GENERALITE          0.50
                        LIVRE NO     1

      INITIATION          0.50
                        LIVRE NO     2

      EXERCICE            0.50
            INITIATION          0.50
                              LIVRE NO     4

            SPECIALITE          0.50
                              LIVRE NO     5

      METHODOLOGIE        0.50
            EXERCICE            0.50
                              LIVRE NO     6

            INITIATION          0.50
                              LIVRE NO     7

      MICROPROGRAMMATIO 0.50
            GENERALITE          0.50
                              LIVRE NO     8

            EXERCICE            0.50
                              LIVRE NO     9

            INITIATION          0.50
                              LIVRE NO    10

      PROFESSEUR          0.50
                        LIVRE NO     3
```


Requête : "J'aimerais savoir comment on programme?"

1er cas

Livres proposés :

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
1	Introduction to computer programming	0.74	1.00	0.50	0.50
2	La programmation systématique	0.74	1.00	0.50	0.50
3	Programming teaching techniques	0.74	1.00	0.50	0.50
8	La microprogrammation	0.74	1.00	0.50	0.50
9	Problèmes de microprogrammation	0.74	1.00	0.50	0.50
10	Microprogramming	0.74	1.00	0.50	0.50
6	Programmation modulaire	0.74	1.00	0.50	0.50
7	Méthodologie	0.74	1.00	0.50	0.50
4	Problems for computer solutions	0.74	1.00	0.50	0.50
5	the art of computer science	0.74	1.00	0.50	0.50

Commentaires :

Les obstacles ne sont pas suffisamment forts pour empêcher le système d'atteindre tous les livres de programmation.

requête : "je recherche un recueil d'exercices pour apprendre la programmation?"

livres proposés :

1er cas

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	apprendre	exercice
4	Problems for computer solutions	0.83	1.00	0.50	1.00
5	The art of computer science	0.83	1.00	0.50	1.00
6	Programmation modulaire	0.83	1.00	0.50	1.00
9	Problèmes de microprogrammation	0.83	1.00	0.50	1.00
1	Introduction to computer programming	0.74	1.00	0.50	0.50
2	La programmation systématique	0.74	1.00	0.50	0.50
3	Programming teaching techniques	0.74	1.00	0.50	0.50
8	La microprogrammation	0.74	1.00	0.50	0.50
10	Microprogramming	0.74	1.00	0.50	0.50
7	Méthodologie	0.74	1.00	0.50	0.50

Commentaires :
Les quatre livres dans lesquels il y a des exercices apparaissent plus pertinents que les autres.

cas n°2 : traitements avec la procédure "programmation" ci-contre et sans procédure stratégique.

```
FONDS DOCUMENTAIRE
******************

PROGRAMMATION      1.00
        GENERALITE          0.80
                          LIVRE NO     1

        INITIATION          1.00
                          LIVRE NO     2

        EXERCICE            1.00
                INITIATION          0.90
                                  LIVRE NO     4

                SPECIALITE          1.00
                                  LIVRE NO     5

        METHODOLOGIE        1.00
                EXERCICE            0.90
                                  LIVRE NO     6

                INITIATION          0.90
                                  LIVRE NO     7

        MICROPROGRAMMATIO   1.00
                GENERALITE          0.90
                                  LIVRE NO     8

                EXERCICE            0.90
                                  LIVRE NO     9

                INITIATION          0.90
                                  LIVRE NO     10

        PROFESSEUR          1.00
                          LIVRE NO     3
```

Requête : "j'aimerais savoir comment on programme?"

Livres proposés :

2ème cas

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Commentaires :

La requête étant évasive, le système propose avec "indulgence" un livre de généralités.

requête : "je recherche un recueil d'exercices pour apprendre la programmation ?"

Livres proposés :

2ème cas

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	apprendre	exercice
4	Problems for computer solutions	0.83	1.00	0.50	1.00
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Commentaires :

Le système répond pertinemment en proposant un livre d'exercices de programmation. (livre n°4)

Procédures documentaires et stratégiques

cas n°3 : il n'y a pas d'obstacle dans les procédures stratégiques.

```
PROGRAMMATION      1.00
        GENERALITE          0.80
                          LIVRE NO    1
        INITIATION          1.00
                          LIVRE NO    2
        EXERCICE            1.00
                INITIATION          0.90
                                  LIVRE NO    4
                SPECIALITE          1.00
                                  LIVRE NO    5
        METHODOLOGIE        1.00
                EXERCICE            0.90
                                  LIVRE NO    6
                INITIATION          0.90
                                  LIVRE NO    7
        MICROPROGRAMMATIO 1.00
                GENERALITE          0.90
                                  LIVRE NO    8
                EXERCICE            0.90
                                  LIVRE NO    9
                INITIATION          0.90
                                  LIVRE NO   10
        PROFESSEUR          1.00
                          LIVRE NO    3
CONNAITRE          0.50
        JUGER               0.50
                RAPPORT             0.80
                CRITIQUE            0.70
                EVALUATION          0.70
        ETUDIER             0.50
                PRECIS              0.50
                        SPECIALITE          0.60
                IMPRECIS            0.50
```


Suite des procédures stratégiques

```
                        INITIATION          0.60
                F1                  0.50
                        IMPRECIS            0.90
                OBJET               0.50
                        DESCRIPTION         0.80
                TRAITEMENT          0.50
                        REALISATION         0.70
                        PERFORMANCE         0.60
                        DESCRIPTION         0.90
        REALISER            0.50
                CONSEILS            0.70
                EXEMPLE             0.70
        F1                  0.50
                ETUDIER             0.90
COMMENT            0.50
        MANIERE             0.90
        MOYEN               0.90
        EXEMPLE             0.90
        ACTIVITE            0.50
                REALISATION         0.80
                CONSEILS            0.90
        UTILISATION         0.50
                MODE D'EMPLOI       0.90
```

requête :"j'aimerais savoir comment on programme?"

3ème cas

Livres proposés :

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
7	Méthodologie	0.8	1.00	0.60	0.75
6	Programmation modulaire	0.8	1.00	0,50	0.86
5	The art of computer science	0.8	1.00	0.60	0.67
4	Problems for computer solutions	0.8	1.00	0.60	0.67
2	La programmation systématique	0.76	1.00	0.60	0.50

Commentaires :
Les deux procédures stratégiques permettent d'associer des objectifs à la requête à travers lesquels le système accède aux documents ci-dessus. Les autres livres ne sont pas atteints car ils ne vérifient pas toutes leurs conditions d'accès. Un fichier inversé ne permet pas toujours de faire ce filtre.

requête : "je recherche un recueil d'exercices pour apprendre la programmation?"

3ème cas

Livres proposés :

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	apprendre	exercice
4	Problems for computer solutions	0.88	1.00	0.60	1.00
5	The art of computer science	0.88	1.00	0.60	1.00
2	La programmation systématique	0.76	1.00	0.60	0.50
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Commentaires : Par rapport aux résultats de la 1re requête, le système ne propose pas de documents de méthodologie (à raison).

Procédures documentaires et stratégiques

cas n°4 : avec des priorités d'accès dans les procédures stratégiques et documentaires.

```
PROGRAMMATION      1.00
        GENERALITE          0.80
                          LIVRE NO     1

        INITIATION          1.00
                          LIVRE NO     2

        EXERCICE            1.00
                INITIATION          0.90
                                  LIVRE NO     4

                SPECIALITE          1.00
                                  LIVRE NO     5

        METHODOLOGIE        1.00
                EXERCICE            0.90
                                  LIVRE NO     6

                INITIATION          0.90
                                  LIVRE NO     7

        MICROPROGRAMMATIO 1.00
                GENERALITE          0.90
                                  LIVRE NO     8

                EXERCICE            0.90
                                  LIVRE NO     9

                INITIATION          0.90
                                  LIVRE NO    10

        PROFESSEUR          1.00
                          LIVRE NO     3

CONNAITRE          1.00
        JUGER               1.00
                RAPPORT             0.80
                CRITIQUE            0.70
                EVALUATION          0.70
        ETUDIER             1.00
                PRECIS              1.00
                        SPECIALITE          0.60
                IMPRECIS            1.00
```


Suite des procédures stratégiques

```
                        INITIATION          0.60
                F1                  0.70
                        IMPRECIS            0.90
                OBJET               1.00
                        DESCRIPTION         0.80
                TRAITEMENT          1.00
                        REALISATION         0.70
                        PERFORMANCE         0.60
                        DESCRIPTION         0.90
        REALISER            1.00
                CONSEILS            0.70
                EXEMPLE             0.70
        F1                  0.70
                ETUDIER             0.90
COMMENT             1.00
        MANIERE             0.90
        MOYEN               0.90
        EXEMPLE             0.90
        ACTIVITE            1.00
                REALISATION         0.80
                CONSEILS            0.90
        UTILISATION         1.00
                MODE D'EMPLOI       0.90
```

Requête : "j'aimerais savoir comment on programme?"

Livres proposés :

4ème cas

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
7	Méthodologie	0.8	1.00	0.60	0.75
6	Programmation modulaire	0.8	1.00	0.50	0.86
4	Problems for computer solutions	0.8	1.00	0.60	0.67
2	La programmation systématique	0.76	1.00	0.60	0.50
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Commentaires : Lorsqu'une requête est imprécise, les chemins dans la procédure stratégique "connaître" dirigent la recherche vers des livres d'initiation (plutôt que des livres de spécialité). La représentation de cette préférence a pour effet de faire disparaître le livre n°5 de la liste des livres qui ont été proposés au cas n°3.

requête :"je recherche un recueil d'exercices pour apprendre la programmation"

Livres proposés :

4ème cas

n° d'ordre	titre des livres	proximité /requête	proximité par rapport aux objectifs de la requête		
			programmation	apprendre	exercice
4	Problems for computer solutions	0.88	1.00	0.60	1.00
2	La programmation systématique	0.76	1.00	0.60	0.50
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Conclusion :

Les résultats obtenus montrent que les procédures stratégiques peuvent permettre au système de faire quelques suppositions sur les requêtes en fonction desquelles la recherche est orientée.

Les cas 3 et 4 indiquent que l'effet recherché peut être positif. Néanmoins, il est certain que ce type d'enrichissement comporte des risques d'erreur. C'est la raison pour laquelle le système ne considère généralement pas très pertinents les documents atteints par ce procédé.

En revanche, il semble se diriger avec davantage "d'assurance" quand la requête est précise : par rapport à la première requête évasive, la portée de la deuxième requête a été moins amplifiée, ce qui s'est traduit par des temps de traitements moins longs ; les livres atteints sont généralement moins nombreux et plus pertinents.

Quant aux plausibilités d'importance, elles jouent différents rôles suivant la valeur qu'on leur accorde:

Si elles sont plus ou moins élevées, elles permettent de privilégier certains accès (cas n°2 et n°4) et de ne pas bloquer les recherches. Dans la réalité, les conditions d'accès ne sont pas du type "tout ou rien". Leur importance doit donc être pondérée pour permettre l'évaluation des différents documents atteints (cas n°4).

ANNEXE 2

Nous voulons vérifier que les différentes sortes de plausibilités élémentaires (plausibilités relationnelles, d'intérêt, d'atténuation, et de dépréciation) ont effectivement le rôle que nous voudrions qu'elles jouent.

Pour effectuer ce test, nous allons analyser les répercussions entraînées par la neutralisation de chacune d'elles, lors du traitement de la requête "j'aimerais savoir comment on programme?" avec les procédures "programmation", "comment" et "connaître".

Auparavant, nous allons traiter la requête sans neutralisation : les résultats serviront de référence.

Requête traitée sans neutralisation

"j'aimerais savoir comment on programme"

AMPLIFICATION DE LA REQUETE

OBJECTIFS	RELATION	PL.PROX /requête	PL.PROX./OBJ.DE LA REQUETE pgmtion	savoir	comment
PROGRAMMATION	VD	1.00	1.00	0.50	0.50
SAVOIR	VD	1.00	0.50	1.00	0.50
COMMENT	VD	1.00	0.50	0.50	1.00
RECHERCHE DOCUMENT					
CONNAITRE	VD	0.86	0.50	0.86	0.50
EXEMPLE	ASS	0.75	0.50	0.50	0.75
MOYEN	ASS	0.75	0.50	0.50	0.75
MANIERE	ASS	0.75	0.50	0.50	0.75
ETUDIER	ASS	0.75	0.50	0.75	0.50
IMPREVU	ASS	0.75	0.50	0.75	0.50
METHODOLOGIE	ASS	0.64	0.50	0.50	0.64
EXERCICE	ASS	0.62	0.50	0.50	0.62
INITIATION	ASS	0.57	0.50	0.57	0.50
RECHERCHE DOCUMENT					

5 LIVRES PROPOSES

Requête traitée sans neutralisation

"j'aimerais savoir comment on programme"

LIVRES PROPOSES

LIVRES		PL.PROX /requête	PL.PROX./OBJ.DE LA REQUETE pgmtion	savoir	comment
LIVRE NUMERO	6	0.80	1.00	0.50	0.74
LIVRE NUMERO	7	0.80	1.00	0.57	0.64
LIVRE NUMERO	4	0.80	1.00	0.57	0.62
LIVRE NUMERO	2	0.75	1.00	0.57	0.50
LIVRE NUMERO	1	0.74	1.00	0.50	0.50

EXECUTION TERMINEE

STPTIME	VIRT	REL	ACT	TRA	CPM	FAULTS	INS	OUTS	RATE-P/S	UC-E	RL-E
2.957	110	77	70	05	73	2550	402	852	14.67	1.44	1.38

R385	STEP	WAIT	CORE	END	DISC	TIME	START	CODE	PROGRAM
	01	00.39	107	19*09*34	26	02.95	19*02*10	TERM	V7SIMULA

TIME	TIME*CORE	CORE-USE	TIME*DISC	DISC-USE
02.95	230.74	95%	857.48	01%

DS I/O-BYTES	DS I/O-CALLS	SHR I/O-BYTES	SHR I/O-CALLS	CR CARDS	CP CARDS	LP PAGES
527685	455	00	00	137	00	32

SELF JOB TERMINATED DAVID 0MIN 19*09*34*

Neutralisation des plausibilités relationnelles

Requête : "j'aimerais savoir comment on programme"

Livres proposés :

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
4	Problems for computer solutions	0.84	1.00	0.60	0.75
7	Méthodologie	0.84	1.00	0.60	0.75
6	Programmation modulaire	0.82	1.00	0.50	0.90
2	La programmation systématique	0.77	1.00	0.60	0.50
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Neutralisation des plausibilités relationnelles

Les relations mises en oeuvre pour atteindre les documents ont été les suivantes :

livre n°4 : 3 relations "associé à "
livre n°7 : 2 "associé à" et une "sorte de"
livre n°6 : 1 "associé à" et 1 "sorte de"
livre n°2 : 1 "associé à "
livre n°1 : 0 ("passage forcé")

En inhibant les plausibilités relationnelles, le système ne sanctionne plus les documents atteints à l'aide de plusieurs relations vagues telles que "associé à".

Neutralisation des plausibilités d'intérêt

Requête : "j'aimerais savoir comment on programme"

Livres proposés :

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
7	Méthodologie	0.90	1.00	0.80	0.75
2	La programmation systématique	0.81	1.00	0.80	0.50
6	Programmation modulaire	0.80	1.00	0.50	0.75
1	Introduction to computer programming	0.74	1.00	0.50	0.50

Neutralisation des plausibilités d'intérêt

Sans neutralisation (page 194) les plausibilités d'intérêt des associations d'objectifs qui permettent l'accès aux documents sont les suivantes :

livre n°7 : 0.9 ; 0.8 ; 0.6
livre n°2 : 0.6
livre n°6 : 0.9 ; 0.8 ; 0.9 ; 0.8
livre n°1 : atteint en forçant le passage.

en supprimant leur effet, le système ne sanctionne plus les documents atteints à l'aide d'associations inopportunes.

Neutralisation des plausibilités d'atténuation

Requête : "j'aimerais savoir comment on programme"

Livres proposés :

n° d'ordre	titre des livres	proximité / requête	proximité par rapport aux objectifs de la requête		
			programmation	savoir	comment
7	Méthodologie	0.53	1.00	0.57	0.64
4	Problems for computer solutions	0.52	1.00	0.57	0.62
6	Programmation modulaire	0.50	1.00	0.50	0.74
1	Introduction to computer programming	0.50	1.00	0.50	0.50
2	La programmation systématique	0.50	1.00	0.57	0.50

Commentaires :

Le nombre d'objectifs satisfaits ne peut atténuer le fait qu'un seul n'ait pu être correctement atteint. Les documents se trouvent alors classés en fonction du plus mauvais objectif.

Une séquence de l'exploration des procédures (sans neutralisation des plausibilités de dépréciation)

```
OBJECTIFOBJET                           71    A ETE ATTEINT    0.99    0.83
          39POINT DE REPRISE CONSULTE
OBJECTIFTRAITEMENT                      39    A ETE ATTEINT    0.99    0.83
          11POINT DE REPRISE CONSULTE
OBJECTIFJUGER                           11    A ETE ATTEINT    0.95    0.73
          28POINT DE REPRISE CONSULTE
OBJECTIFREALISER                        28    A ETE ATTEINT    0.95    0.73
          41POINT DE REPRISE CONSULTE
OBJECTIFUTILISATION                     41    A ETE ATTEINT    1.00    0.69
          23POINT DE REPRISE CONSULTE
OBJECTIFACTIVITE                        23    A ETE ATTEINT    1.00    0.69
          12POINT DE REPRISE CONSULTE
OBJECTIFGENERALITE                      12    A ETE ATTEINT    1.00    0.66
           3POINT DE REPRISE CONSULTE
OBJECTIFINITIATION                       3    A ETE ATTEINT    1.00    0.66
          67POINT DE REPRISE CONSULTE
OBJECTIFMICROPROGRAMMATIO               67    A ETE ATTEINT    1.00    0.66
           6POINT DE REPRISE CONSULTE
OBJECTIFPROFESSEUR                       6    A ETE ATTEINT    1.00    0.66
      *****************
      2 LIVRES PROPOSES
      *****************
LANCEMENT DE LA PROCEDURE DEBLOCAGE
           3POINT DE REPRISE CONSULTE
   9   0
INITIATION
   9   0
LANCEMENT DE LA PROCEDURE DEBLOCAGE
          39POINT DE REPRISE CONSULTE
OBJECTIFTRAITEMENT                      39    A ETE ATTEINT    0.99    0.77
          71POINT DE REPRISE CONSULTE

OBJECTIFOBJET                           71    A ETE ATTEINT    0.99    0.77
          42POINT DE REPRISE CONSULTE
OBJECTIFPRECIS                          42    A ETE ATTEINT    0.99    0.77
          28POINT DE REPRISE CONSULTE
OBJECTIFREALISER                        28    A ETE ATTEINT    0.95    0.69
          11POINT DE REPRISE CONSULTE
OBJECTIFJUGER                           11    A ETE ATTEINT    0.95    0.69
          23POINT DE REPRISE CONSULTE
OBJECTIFACTIVITE                        23    A ETE ATTEINT    1.00    0.66
          41POINT DE REPRISE CONSULTE
OBJECTIFUTILISATION                     41    A ETE ATTEINT    1.00    0.66
           6POINT DE REPRISE CONSULTE
OBJECTIFPROFESSEUR                       6    A ETE ATTEINT    1.00    0.64
          67POINT DE REPRISE CONSULTE
OBJECTIFMICROPROGRAMMATIO               67    A ETE ATTEINT    1.00    0.64
           3POINT DE REPRISE CONSULTE
INITIATION                   3  A UN RAPPORT AVEC LA REQUETE
LIVRE NUMERO       1013   A ETE ATTEINT      1.00      1.00
        1013POINT DE REPRISE CONSULTE
          43PROPOSITION D UN LIVRE
          12POINT DE REPRISE CONSULTE
OBJECTIFGENERALITE                      12    A ETE ATTEINT    1.00    0.64
      *****************
      3 LIVRES PROPOSES
      *****************
```

colonne des plausibilités de proximité/requête

colonne des plausibilités de dépréciation

Chaque noeud atteint est estimé puis classé en fonction de sa plausibilité de priorité calculée par (plausibilité de proximité du chemin menant au noeud/requête x plausibilité de dépréciation du noeud)

Neutralisation des plausibilités d'atténuation

Il faudrait un plus grand nombre de documents pour évaluer l'effet de ces plausibilités sur les résultats de la recherche documentaire.

En les neutralisant, nous avons remarqué un gain de temps appréciable qui s'explique par la suppression des réordonnancements des points de reprise (cf. ordre de priorité des questions page 125). Un compromis serait à trouver car dans la version actuelle, ces réordonnancements semblent trop fréquents : ils sont faits systématiquement, chaque fois que la réponse à une question atteinte est indéterminée.

GLOSSAIRE

- *chemin (page 85) :*
  suite de questions apparaissant consécutivement.

- *chemin idéal (page 113) :*
  liste des objectifs qui indexent une requête et qui sont mis sous la forme d'un chemin.

- *domaine de validité (page 64) :*
  il doit se rapporter suffisamment à la requête pour que le lien sémantique dans lequel il figure soit valide.

- *lexique (page 36) :*
  ensemble des objectifs connus a priori par le système documentaire.

- *objectif (page 31) :*
  tout terme du langage naturel appartenant au lexique et pouvant être utilisé pour décrire les requêtes, les lecteurs, les documents...

- *plausibilité d'atténuation (page 114) :*
  elle est calculée en fonction du nombre d'objectifs d'une requête. Lors de la comparaison d'un chemin d'un questionnaire avec la requête, elle atténue la répercussion sur le résultat cherché d'un éloignement sémantique entre le chemin et un des objectifs de la requête.

- *plausibilité de dépréciation (page 126) :*
  plausibilité associée à un noeud atteint dans un questionnaire et qui décroît chaque fois qu'une tentative pour répondre à la question du noeud se solde par une indétermination.

- *plausibilité d'importance (page 85) :*
  valeur représentant l'importance des questions des procédures stratégiques et documentaires.

- *plausibilité d'intérêt (page 62) :*
  valeur indiquant l'opportunité d'associer un objectif à un autre

- *plausibilité de priorité (page 125) :*
  elles permettent d'ordonner les questions atteintes dans les procédures stratégiques ou documentaires en fonction des succès remportés par la recherche pour atteindre ces questions et en fonction de leur plausibilité de dépréciation.

- *plausibilité de proximité (page 58) :*
  fonction donnant une évaluation de la "proximité sémantique" entre ses deux arguments.

- *plausibilité relationnelle (page 61) :*
  représente les préférences accordées aux différents types de relation.

- *procédure documentaire (page 95) :*
  questionnaire représentant les chemins d'accès aux documents.

- *procédure stratégique (page 82) :*
  questionnaire destiné à l'enrichissement des requêtes grâce à l'évaluation de l'esprit dans lequel elles doivent être traitées.

- *procédure structurale (page 139) :*
  procédure à l'aide de laquelle le système peut répondre à une question en examinant les relations entre les objectifs de la requête.

- *série(s) (page 59) :*
  loi de composition de deux plausibilités qui affaiblit la plausibilité résultante.

- *seuil d'amplification (ou seuil de proximité) (page 75) :*
  seuil paramétré indiquant l'éloignement sémantique maximale autorisé pour effectuer une association.

- *thème (page 40) :*
  désigne l'objectif le plus important de la requête.

- *parallèle (//) (page 60) :*
  loi de composition de deux plausibilités qui renforce la plausibilité résultante.

- *thésaurus (page 55) :*
  ensemble de liens sémantiques servant à amplifier la portée des requêtes.

BIBLIOGRAPHIE

I. INTELLIGENCE ARTIFICIELLE, COMPRÉHENSION DU LANGAGE NATUREL

1 AKNIN Y., BONNET A., COULON D., JAKOB F., KAYSER D., LANCEL J.M., MONFILS M. :
Knowledge représentation and approximate reasoning for natural language comprehension, Centre de Recherche en Informatique de Nancy, n° 77.R.016, 1977.

2 BAM WISHO : La compréhension des textes écrits, rapport P 69.05, juillet 1969, Service de la recherche pédagogique, Genève.

3 CHARNIAK E., WILKS Y. : Computational semantics, Fundamental studies in computer science, vol.4, North-Holland, 1976.

4 COULON D., KAYSER D. : Plausibility as an evaluation in knowledge representation, Article , Centre de Recherche en Informatique de Nancy, Ecole des Mines,54042 Nancy Cedex.

5 COULON D., KAYSER D., BONNET A., LANCEL J.M. MONFILS M. :
Essai de compréhension d'un texte à l'aide d'un réseau sémantique de procédures, Congrès AFCET Gif/Yvette, CRIN 76.R.039 (Nancy), Nov.1976.

6 COULON D., KAYSER D., BONNET A., LANCEL J.M., MONFILS M. :
Description générale d'un système de réponse aux questions, 3 rapports de contrats SESORI (N° 76162 et 76173), Avril 1977, Janvier 1978, Juin 1978, Université de Paris Sud, Centre d'Orsay 91405 Orsay.

7 COULON D., KAYSER D., FUSS J., JAKOB F., MONFILS M. :
Premier bilan d'une expérience de compréhension du langage naturel, AFCET 1979, Intelligence Artificielle, CRIN, Ecole des Mines, Parc de Saurupt, 54042 Nancy cedex, France.

8 KAYSER D. :
Vers une modélisation du raisonnement "approximatif", Colloque sur "la représentation des connaissances et le raisonnement dans les sciences de l'homme et de la société", Saint-Maximin, Sept.1979.

9 SCHANK R.C. :
Conceptual information processing, Fundamental studies in computer science, vol.3, North-Holland, 1975.

10 WINOGRAD T. :
Understanding natural language in cognitive psychology, vol.3 n°1, pp.1-191, New-York, Academic-Press, Janvier 1972.

10b IRIA-SESORI :
La compréhension (acquisition, représentation et utilisation des connaissances), Arc et Senans, 17-18 mai 1977.

10c DEBILI F. :
Une méthode de construction automatique de familles de mots, Centre d'Etudes Nucléaires de Saclay , Juillet 1979.

10d ANDREEWSKY A., COMBRISSON F., FLUHR C.
Le problème de l'identification automatique des concepts, Centre d'Etudes Nucléaire de Saclay, Novembre 1975.

10e FILLMORE
The case for case. Universals in linguistic theory, E.Bach & R.T. Harms, ed., Chicago, 1968.

10f MINSKY M.
A framework for representing knowledge, AI, 1974.

10g KUNO
The predicture analyser and a path elimination technique, Communications of the association for computing machinery, 1965.

II. Classologie - Informatique Documentaire

11 ALOUCHE F., BELLY N., CROS R.C., GARDIN J.C., LEVY F., PERRIAULT J. :
Economie générale d'une chaîne documentaire mécanisée, Gauthier-Villars, Paris, 1967.

12 BASSANO J.C. :
Acquisition de connaissances à partir de textes : une expérimentation en documentation automatique, rapport de recherche n°41, Laboratoire de recherche en Informatique, Université de Paris-Sud, Centre d'Orsay, Mai 1979.

13 BASSANO J.C. :
Conceptualisation automatique appliquée à l'interrogation des bases de données bibliographiques, Rapport de recherche n°49, Laboratoire de recherche en Informatique,Université de Paris-Sud, Centre d'Orsay, Juillet 1979.

14 BASSANO J.C. :
Logiciel autoadaptatif pour un réseau de systèmes documentaires corrélés, rapport de recherche n°50, Laboratoire de Recherche en Informatique, Université de Paris-Sud, Centre d'Orsay, Sept.79.

15 BELLY N., BORILLO A., VIRBEL J., SIOT-DECAUVILLE N. :
Procédures d'analyse sémantique appliquées à la documentation scientifique, Gauthier-Villars, Paris, 1970.

16 BORRELLY L., CHOURAQUI E. :
Le système documentaire Satin 1: Description générale et manuel d'utilisation CNRS, Paris,1974.

17 CHAUMIER J. :
L'analyse documentaire : le traitement linguistique et l'information documentaire, Entreprise Moderne d'édition, Paris 1977.

18 CII-HB :
Mistral III/Mistral IV : Guide de mise en oeuvre, réf : OOF3 7511 REVO, Centre de documentation France CEDOC, 68 route de Versailles, 78430 Louveciennes, 1978.

19 CII-HB :
Mistral III/Mistral IV : Guide de l'utilisateur, réf : OOF3 74 66 REVO, Centre de documentation France CEDOC, 68 route de Versailles, 78430 Louveciennes, 1978.

20 CROS R.C., GARDIN J.C., LEVY F. :
L'automatisation des recherches documentaires, Un modèle général : le Syntol, Gauthier-Villars, Paris, 1968.

21 DESVALS Hélène :
Comment organiser sa documentation scientifique?,Gauthier-Villars, Paris, 1975.

22 DOBROWOLSKI Z. :
Etude sur la construction des systèmes de classification, Gauthier-Villars, Paris, 1964.

23 DUBUC R. :
La classification décimale universelle : manuel pratique d'utilisation, documentation et information, Paris, 1965.

24 GUILLOUX R. :
Réseaux et système de documentation, Gauthier-Villars, Paris, 1975.

25 IRIA (Ecole de l'IRIA) :
Support de cours "Informatique et histoire", conférenciers: CHOURAQUI E., FOSSIER L., GENET J.Ph., GUENOCHE A., KLAPISCH C., METMAN J., TOURNIER M., VIRBEL J., ZARRI G.P., Maison des Sciences de l'Homme, paris, Octobre 1979.

26 ISSAAC E. :
Automatisation du prêt dans les bibliothèques de Grenoble, Article écrit en 1976.

27 SALTON G. :
Automatic information organization and retrieval, McGraw-Hill, Inc, New-York, 1968.

28 SALTON G. :
The Smart retrieval system : experiments in automatic document processing, Prentice-Hall Inc, Englewood Cliffs (New Jersey), 1971.

29 THILL A. :
Les systèmes de prêt automatisés dans les bibliothèques publiques françaises, Journées d'étude sur les systèmes automatisés de prêt dans les bibliothèques (article), octobre 1976.

30 VAN SLYPE Georges :
Conception et gestion des systèmes documentaires, Les Editions d'Organisation, Paris, 1977.

31 VICKERY B.C. :
La classification à facettes : guide pour la construction et l'utilisation de schémas spéciaux, Gauthier-Villars, Paris,1963.

32 ZARRI G.P. :
Projet Reseda/1, Rapport sur les recherches effectuées du 01 Octobre 1978 au 01 Avril 1979, CNRS-IRIA, LISH (Laboratoire d'Informatique pour les Sciences de l'Homme, 54 bv Raspail, 75270 Paris Cedex 06).

33 Journées de travail St-Vallier :
Les systèmes d'information documentaire : thèmes de recherche actuels, 1978.

34 NAULLEAU D.
Thésaurus conversationnel. Conception et réalisation d'un outil aidant à l'indexation et à la formulation de questions, Thèse de 3ème cycle, Paris VI, 1973.

35 BASSANO J.C.
Base de données textuelles en documentation automatique.
Rapport de recherche n°75, Université de Paris Sud, Centre d'Orsay, LRI, Bât. 490, 91405 Orsay, Juin 1980.

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

AUTORISATION DE SOUTENANCE DE THESE DE DOCTORAT DE 3ème CYCLE

VU LE RAPPORT ETABLI PAR :

Monsieur le Professeur COULON Daniel

le Président de l'Institut National Polytechnique de Lorraine autorise :

Monsieur DAVID Jean-Marie

à soutenir, devant l'I.N.P.L., une thèse de Doctorat intitulée :

"SIMULATION DE L'ACTIVITE D'UN DOCUMENTALISTE"

en vue de l'obtention du titre de DOCTEUR DE TROISIEME CYCLE - SPECIALITE "INFORMATIQUE"

Fait à NANCY, le 18 Novembre 1980

Le Président de l'I.N.P.L.

C. PAIR