Université de Montréal

Extraction automatique de filtres dans le cadre de la production automatique de résumés

par

Mazen Tout

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

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

Avril, 2001

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Extraction automatique de filtres dans le cadre de la production automatique de résumés

présenté par :

Mazen Tout

a été évalué par un jury composé des personnes suivantes :

LAPALME, Guy : Directeur de Recherche

BENJIO, Yoshua : Président Rapporteur

KEGL, Balázs : Membre de Jurée

Mémoire accepté le : 9 octobre 2001

Sommaire

La production de résumés automatiques est une tâche bien connue dans les études de la langue naturelle. Avec les nouveaux concepts et recherches en intelligence artificielle, l'introduction de techniques d'apprentissage dans la langue naturelle est un important objet d'intérêt. En particulier, la notion de découverte de patrons et de formes dans l'interprétation de la langue en général, et dans le domaine de la production de résumé plus spécifiquement.

Notre mémoire étudie un système de production de résumé automatique ***SumUm*** qui utilise une méthode appelée la sélection indicative. Nous en étudions les méthodes d'interprétation et d'extraction de phrase ainsi que le choix manuel de patrons. Le but du mémoire est de concevoir, à l'aide de principes d'intelligence artificielle et d'algorithmes d'apprentissage, une méthode de découverte automatique de patrons afin de diminuer la dépendance de ***SumUm*** à un domaine spécifique.

Nous avons développé un algorithme de découverte de patrons, ***GlobSum,*** basé sur plusieurs caractéristiques rencontrées dans les méthodes et algorithmes d'apprentissage. L'algorithme débute par une phase d'entraînement sur un corpus, suivi d'un algorithme de découverte et de filtrage qui dégage automatiquement les patrons. Ces patrons sont ensuite incorporés à ***SumUm***.

Nous avons évalué l'effet de l'incorporation de ***GlobSum*** à ***SumUm*** qui n'a entraîné qu'une faible détérioration de performance dans ***SumUm*** démontrant ainsi la viabilité de ***GlobSum*** pour l'identification automatique de patrons.

Mots clés : Découverte de patrons, résumé automatique, apprentissage.

Table de matières

1 Introduction 1

1.1 La limitation des résumés 2

1.2 Plan du mémoire 4

2 Les résumés automatiques, « Analyse Sélective » dans SumUM 6

2.1 Les résumés automatiques 7

2.1.1 Extraction de phrases 8

2.2 L'analyse sélective 11

2.3 SumUM 14

2.3.1 L'interprétation dans SumUm 14

2.3.2 Le dictionnaire conceptuel et les patrons 18

2.3.3 Sélection indicative, un remplissage de template 22

2.4 Limitations de SumUM 27

2.4.1 Le choix de patrons 28

2.5 Besoin d'apprentissage 29

3 L'apprentissage informatique 30

3.1 L'histoire de l'apprentissage 31

3.2 Les différentes classes d'apprentissage 32

3.3 GlobSum, un algorithme de découverte supervisé 51

4 Intégration de l'algorithme GlobSum dans SumUM 53

4.1 Comportement de *SumUm* sur les documents non techniques 54

4.2 Structure de GlobSum 57

5 Évaluations de patrons et de qualité de GlobSum 76

6 Conclusions et amélioration futures 83

Liste de tableaux

Table 1: Catégories Lexicales 16
Table 2: Un fragment d'une phrase interprétée au niveau syntaxique et lexical 16
Table 3: Distribution de l'information 19
Table 4: Liste de quelques concepts 20
Table 5: Liste de relations 20
Table 6: Le template Possible Topic 23
Table 7: Patron des types d'information 24
Table 8: Exemple de l'histoire de crédit 45
Table 9: Précision et rappel sur les documents non techniques 55
Table 10: Jugements de la qualité et degré indicatif pour les documents non techniques 56
Table 11: Table de connaissances d'attributs 64
Table 12: Table de groupes et de patrons 70
Table 13: Rappel et précision des documents techniques par SumUm et GlobSum 77
Table 14: Changement du rappel et de la précision avec la variation du "seuil".... 80

Liste de figures

Figure 1: L'analyse sélective 12
Figure 2: L'interprétation dans SumUm 15
Figure 3: Le dictionnaire conceptuel 22
Figure 4: Sélection et Remplissage de Template 25
Figure 5: Un résumé de SumUm 26
Figure 6: Changement de la performance d'un état par l'introduction de nouvelles expériences 30
Figure 7: L'algorithme Find-S 34
Figure 8: Un espace de concept, du général (plus haut) au spécifique (plus bas) ... 35
Figure 9: L'algorithme de recherche de version d'espace du spécifique au général 36
Figure 10: Le résultat de l'application de l'algorithme de la figure 9 sur l'espace de la figure 8 37
Figure 11: L'algorithme de recherche de version d'espace du général au spécifique 38
Figure 12: Le résultat de l'application de l'algorithme de la figure 11 sur l'espace de la figure 8 39
Figure 13: l'algorithme d'élimination de candidat 41
Figure 14: Le résultat de l'application de l'algorithme 11 sur l'espace de la figure 8 42
Figure 15: l'algorithme ID3 44
Figure 16: Arbre de décision pour la classification de crédit 46
Figure 17: Formation de catégories 50
Figure 18: La structure de l'algorithme de découverte de patrons 60
Figure 19: Le tagger Lexical 60
Figure 20: L'interprétation syntaxique 61
Figure 21: Interprétation sémantique 62
Figure 22: L'algorithme de groupage 68
Figure 23: L'intégration de SumUm et GlobSum 73

Figure 24: Interface de GlobSum et SumUM 74
Figure 25: Variation de Précision de GlobSum 79
Figure 26: Variation de Rappel de GlobSum 79
Figure 27: Variation du rappel à 3%, 6% et 10% 81
Figure 28: Variation de la précision à 3%, 6% et 10% 82

Chapitre 1

1 Introduction

Les résumés automatiques des textes, forment de plus en plus, un problème pratique pressant. Avec les nouveaux développements dans le domaine du traitement des textes, avec la croissance rapide de la disponibilité des livres et encyclopédies sur CD, et surtout avec les développements exponentiels dans « *l'Internet* », le volume des textes « *électroniques* » est devenu pratiquement impossible à manipuler sans une utilisation massive d'outils d'extraction et de filtrage et les ressources attribuées pour les recherches dans la production des résumés automatiques ont multiplié en magnitude (*Tait, 1983*). Par conséquent, les utilisateurs sont confrontés à une vaste quantité de textes, souvent très difficiles non seulement à absorber, mais aussi à parcourir. La production des résumés automatiques peut aider à résoudre ce problème. L'abstraction et la compression offerte par un résumé résoudront ainsi les difficultés de couverture de documents et de repérage d'informations pertinentes.

Un résumé doit contenir l'essentiel d'un texte source. Une consultation du résumé devrait permettre de juger la pertinence du document par rapport à l'information recherchée (*Maizell et al. 1971*). Le lecteur a alors moins souvent besoin de lire le document. La problématique de la production automatique des résumés a connu divers changements au cours des années. Les développements informatiques ont permis le développement de nouvelles méthodes dans le traitement du langage naturel englobant les couches lexicales, syntaxiques et sémantiques. Cette puissance de calcul a fait avancer les algorithmes de recherche, de filtrage et d'extraction, mais aussi l'intelligence artificielle, en particulier pour les applications de traitement des textes à l'aide de l'apprentissage automatique.
L'intelligence artificielle est de plus en plus intégrée aux traitements des textes. Les études statistiques, les règles de production ou les règles inductives sont souvent incorporées pour compléter les traitements (*Maybury, 1995*). La production des

résumés automatiques profite à son tour, des bénéfices de la fusion entre l'intelligence artificielle et le traitement du langage.

1.1 La limitation des résumés

Un résumé est un texte qui présente le contenu *essentiel* d'un texte source. Une utilisation possible est la détermination de la *pertinence* d'un document lors d'une recherche d'information sans besoin de consultation complète du document source en conservant la cohérence, la couverture et l'inclusion des événements principaux (*Alterman, 1991*).

La production des résumés est un problème difficile du traitement de la langue naturelle puisque, pour le faire correctement, on doit comprendre le but d'un texte. Ceci exige une analyse sémantique, le traitement du discours, et l'interprétation déductive (détermination de contenu en fonction des connaissances globales du monde). Cette dernière étape, en particulier, est complexe, parce que les systèmes ne peuvent l'accomplir sans beaucoup de connaissance du monde. Par conséquent, plusieurs tentatives d'effectuer de vraies abstractions n'ont pas été très réussies.

Il faut arriver à un compromis acceptable entre la profondeur de l'analyse et la robustesse à différents types de textes. Lorsque les systèmes analysent et interprètent les données assez profondément pour produire des sommaires d'une très bonne qualité, ils restent le plus souvent limités à certains domaines d'applications (*Maizell et al..,1971*). Lorsqu'ils fonctionnent avec plus de robustesse couvrant plus de texte avec moins de restrictions de domaine d'application, leur analyse n'est pas assez profonde pour transformer les données d'entrée en un vrai sommaire, on se limite alors à l'extraction de sujets (topic extraction).

D'un côté, les techniques symboliques, telles que l'utilisation des programmes d'analyse syntaxique, les grammaires, et les représentations sémantiques, n'arrivent pas à traiter efficacement de grandes masses de textes (*Allen,1994*). D'un autre coté, la recherche documentaire et d'autres techniques statistiques, basées sur le

comptage et le groupement de mots, ne peuvent pas créer des vrais sommaires. Ces méthodes ne traitent que la surface du mot et non pas le niveau conceptuel. La plupart des techniques existantes de production de résumés automatiques n'essayent même pas l'analyse superficielle. Elles sont non adaptatives, elles ne peuvent pas apprendre d'exemples des mots clés convenablement choisis et de bons sommaires. Il est pourtant envisageable de penser que la tâche de production d'un sommaire automatique pourrait profiter de l'utilisation des algorithmes d'apprentissage et des techniques linguistiques. Le but final d'un résumé est d'établir un système autonome qui extrait les concepts centraux à partir d'un texte.

Notre intérêt se trouve dans l'étude de système de production de résumé complexe. L'une des tâches les plus coûteuse dans ces systèmes est la détermination des patrons. Jusqu'à maintenant, cette identification a été accomplie à la main par appariement et comparaison, une tâche qui non seulement prend beaucoup de temps mais aussi beaucoup d'effort et de complexité. On débute normalement par une étude de corpus comprenant un triage, un alignement et une comparaison d'informations (*Riloff E, 1993*). Plus d'information sur les alignements de corpus se trouve dans (*Teufel, Moens 1997*). Notre objectif est l'ajout d'un algorithme d'apprentissage dans le contexte de la production des résumés automatiques. Plus spécifiquement, étudier un système de production de résumé automatique complexe mais spécialisé pour les documents techniques, ***SumUm*** qui a été développé à l'Université de Montreal par Horacio Saggion (*Saggion, 2000d*) dans sa thèse de PhD. ***SumUm*** utilise, dans une grande partie de ces phases d'interprétation et sélection, des patrons qui ont été identifiés à la main par une étude d'un corpus de documents techniques. On désire développer une méthode automatique d'extraction de ces « *patrons* » lexicaux afin de diminuer la restriction de ***SumUm*** aux domaines techniques.

Un document dans un domaine spécifique utilise un ton, une voix, des mots et des règles régulières et différentes des documents d'un autre domaine. Par exemple, les mots « *bombe* », « *terroriste* », « *otage* » à une voix active ont une grande probabilité d'être répétés dans des articles sur les actes terroristes, alors que les

mots « *imprimante* », « *papier* », « *mémoire* », « *démarrage* » et « *disque dur* » apparaîtront beaucoup plus souvent dans le domaine des ordinateurs. Il est donc raisonnable de supposer que ces répétitions sont caractéristiques de chaque domaine et peuvent ainsi être découvertes automatiquement par un algorithme de découverte de patrons, le but que nous essayerons d'accomplir.

La majeure partie de l'interprétation dans ***SumUm*** a été obtenue suite à une étude et une comparaison d'un grand corpus (100-120 documents) duquel ***Horacio Saggion*** (*Saggion, Lapalme 1998*) (*Saggion, 2000c*) a extrait des concepts et des relations et les termes lexicaux correspondant.

Nous allons montrer dans ce travail comment cette tâche d'extraction pourrait être automatisée ce qui faciliterait le passage d'un domaine à un autre. ***SumUm*** sera intégré à un nouveau système qui dégagera automatiquement les patrons: ***GlobSum***. Peut être le but de produire un système de production de résumé automatique complexe et puissant pour « *tous les domaines* » sera plus facile.

1.2 Plan du mémoire

Le mémoire est divisé en 6 chapitres. Dans le chapitre 2, nous présenterons le processus général de la production des résumés automatiques, en particulier la sélection indicative utilisée dans ***SumUm***. Nous détaillons ensuite deux phases d'interprétation et de sélection de ***SumUm*** et le choix de patron afin d'en faire ressortir les limitations et le besoin d'apprentissage.

Le Chapitre 3 présente les aspects de l'apprentissage automatique pertinents à notre problème : L'histoire, les différents types et plus spécifiquement, l'apprentissage supervisé de découverte et de « *reconnaissance de patron* ». Nous expliquerons aussi quels fragments et caractéristiques de ces algorithmes ont aidé dans le développement de notre algorithme.

Le chapitre 4 présente ***GlobSum*** et l'intégration de l'algorithme choisi dans ***SumUm***, ses caractéristiques, ses catégories et ses attributs, ainsi que ses limites.

Dans le chapitre 5, une évaluation de ***GlobSum*** est décrite. Les conclusions sont présentées au chapitre 6.

Chapitre 2

2 Les résumés automatiques, « Analyse Sélective » dans SumUM

Le rapport entre les processus de compréhension de langage naturel et la production automatique de résumé a été étudié en science cognitive, linguistique et intelligence artificielle. L'idée n'est pas nouvelle, le premier système de résumé date des années 50. La plupart des résumeurs utilisent en majorité des méthodes statistiques, surtout dans les modèles d'analyses de phrases (Parsing models) (*Charniak, 1993*) mais elles ont profité dans les années 80 de l'introduction des nouvelles idées d'intelligence artificielle. Aujourd'hui, les applications combinent ces deux méthodes pour un meilleur résultat. La tâche de produire des résumés automatiques est complexe. Une compréhension du texte source est nécessaire, ainsi qu'une interprétation sémantique et une réformulation. Cette tâche exige des compétences linguistiques, syntaxiques et sémantiques présentes chez les humains mais qui sont difficiles à implanter de manière automatique. L'identification des différents processus cognitifs dans chaque étape de la composition d'un résumé est importante (*Cremmins,1982*). Il existe deux types de systèmes de production de résumés : Des « *extracteurs* » de phrases qui permettent d'arriver rapidement à un résumé simple, rapide et ce dans plusieurs domaines. Ces extracteurs sont surtout utilisés sur le Web (*Cooley, Mobasher, Srivastava, 1997*). Le degré de complexité et les méthodes d'extraction de phrases sont simples. Le traitement se limite au niveau lexical et parfois à une légère analyse syntaxique. L'autre type de résumeur est plus complexe et plus détaillé. Il utilise plusieurs méthodes d'interprétation et d'extraction de phrases. Par exemple des analyses lexicales, syntaxiques et même sémantiques (*Alterman, 1991*). Il ne peut toutefois être appliqué qu'à des domaines spécifiques. Nous nous intéressons à ce second type de résumé. Les résumés détaillés, plus pertinents mais qui peuvent souffrir d'un manque de portabilité et de flexibilité à cause de la limitation à un domaine. Nous allons toutefois essayer d'éliminer la dépendance par rapport à un domaine à l'aide d'apprentissage automatique. Nous allons nous appuyer sur un programme de production de résumé

automatique, ***SumUm*** développé par ***Horacio Saggion*** (*Saggion, 2000d*), et l'augmenter avec une méthode d'apprentissage de patrons afin de faciliter le passage de ***SumUm*** d'un domaine à l'autre pour arriver au nouveau système ***GlobSum***.

Ce chapitre étudiera le système ***SumUm*** et l'analyse sélective, la méthode qu'il utilise dans la production de résumé, tout en détaillant la formation manuelle des patrons qu'il utilise. Nous présentons les méthodes d'interprétation dans la production des résumés automatiques et nous décrirons les différents niveaux d'analyse lexicale, syntaxique et sémantique. On introduira ensuite, l'analyse sélective une méthode pour la production de résumé automatique, suivie par l'application de cette méthode sur ***SumUm,*** comment la détermination des patrons est faite et comment l'automatisation de cette phase par un algorithme d'apprentissage diminuera la dépendance de ***SumUm*** au domaine technique.

2.1 Les résumés automatiques

Le processus de production automatique de résumés comporte une ou plusieurs de ces étapes dépendamment à sa qualité :

1. ***Segmentation du texte source en unités*** : Phrases, paragraphes, titres, sections, bibliographie, notes et diagrammes.
2. ***Interprétation syntaxique et conceptuelle*** : Utilisation de grammaires et d'automates pour les analyses syntaxiques.
3. ***Sélection du contenu le plus représentatif*** : Les mots, phrases et sections les plus pertinents.
4. ***Condensation, élimination et généralisation*** : Élimination des répétitions, acronymes, des références sans antécédents…
5. ***Génération*** : Production d'un nouveau texte à partir de plusieurs transformations spécifiques.

La segmentation du texte est souvent accomplie par un programme externe indépendant. Le niveau de complexité de l'interprétation ajoute à son tour à la qualité du résumé : L'extraction de phrase est la majeure partie de l'interprétation.

Nous présentons dans les prochaines sections les différentes méthodes d'extraction de phrases.

2.1.1 Extraction de phrases

La phase d'extraction de phrases a comme objectif de localiser dans le texte source les phrases les plus pertinentes. Elle s'occupe de l'isolation des fragments de texte pertinents dans le document source, suivie de l'extraction de l'information importante de ces fragments et la formation d'un espace cohérent englobant ces informations (*Cowie and Lehnert, 1996*). Plusieurs méthodes ont été et sont encore utilisées :

1. ***Distribution de termes*** : L'idée est de considérer « *importantes* » les phrases contenant des mots pertinents du texte. Un mot est pertinent si sa fréquence est élevée sans être un des mots communs dans le langage traité. Le texte source est analysé pour calculer les fréquences des différents mots, transformant les mots en leurs racines comme « *manger* », « *mangé* » et « *mangeait* ». Une liste dynamique des mots courants est formée selon le domaine traité : Par exemple, le mot « *diagnostic* » est commun dans le domaine de la « *médecine* ». La liste de fréquences est triée pour former la liste de distribution de termes. Le poids d'une phrase est alors déterminé en utilisant le texte source et la liste de fréquences. Plusieurs méthodes sont utilisées pour calculer ces poids : La somme des fréquences des mots de la phrase, la cooccurrence de certains termes, la proximité des mots ou bien la fréquence de terme accouplée à l'inverse de la fréquence des documents, le tf-idf (*Salton, 1994*). Les phrases les plus « *pesantes* » sont choisies pour le résumé final. Cette méthode d'extraction est utilisée dans tous les systèmes de production des résumés automatiques, elle est majeure dans les résumeurs simples puisqu'elle offre un critère de pertinence indépendant du domaine, alors qu'elle est utilisée d'une façon complémentaire avec d'autres méthodes d'extraction dans les résumeurs avancés. Plusieurs méthodes de mesures de fréquence existent comme le tf-idf ou le TLTF qui multiplient une fonction monotone de la longueur du terme par une fonction

monotone de la fréquence du terme (Term Length, Term Frequency) introduit par Kantrowitz.

2. ***Les phrases «importantes », les sections, positions, le titre et la conclusion*** : L'idée est d'améliorer la méthode de fréquence afin de raffiner la sélection des phrases. Ces méthodes complètent la méthode de distribution de fréquences. La méthode de position accorde plus d'importance à la première et à la dernière phrase d'un paragraphe et surtout à l'introduction et à la conclusion (*Lin, Hovey, 1997*). Ces phrases, considérées thématiques, indiquent le contenu de leur paragraphe correspondant. La méthode de phrases « *importantes* » (Cue Words) (*Edmundson H.P., 1969*) prend en considération certains mots clé pour augmenter l'importance d'une phrase. Ex. : « *ce document présente XXX* » est considérée comme importante quelque soit « *XXX* ». Cette méthode considère donc les phrases qui contiennent les mots qui marquent des informations importantes. La méthode « *titre et conclusion* » considère les mots qui se répètent dans les titres et les conclusions du document source. Cette méthode doit tenir compte du problème d'ambiguïté dans le cas où les résumés sont généraux et non spécifiques au domaine. Un titre sarcastique, une métaphore peut confondre cette méthode. Les résumeurs utilisent donc souvent une combinaison de ces méthodes afin d'améliorer les résultats. La méthode des phrases importante devient de plus en plus utilisées dans les système de production de résumé automatique. Plus d'information sur ce sujet se trouve dans (*Boguraev, Kennedy, 1997*).
3. ***Processus indicatif*** : Le processus est basé sur l'extraction de certaines expressions qui font référence à des concepts et relations. Un dictionnaire ou un index conceptuel sont construits pour grouper les phrases selon leurs traits. Une phrase contenant les mots « *Nous concluons* », qui correspond au concept « *conclusion* » sera souvent considérée pertinente et apparaîtra dans l'extrait résumeur. Il faut noter que la méthode indicative doit être accompagnée par une étude syntaxique pour diminuer les ambiguïtés. Le principe des études indicative a été introduit par Paice (*Paice 1990*).

4. ***Thesaurus*** : Cette méthode est basée sur la formulation de catégories conceptuelles. Un thesaurus d'expressions est formé pour caractériser chaque catégorie. Une première approche attribue des poids aux concepts et sélectionne les phrases avec les concepts les plus importants. La seconde approche, plus complexe, forme des règles sémantiques contextuelles appliquées à la phrase. Ces règles sont souvent la cooccurrence de certains concepts, verbes, genre et temps. Ex. : « *il est nécessaire de montrer* » contient le concept « *démonstration* » avec le verbe « *être* » et l'adjectif « *nécessaire* », cette combinaison de marqueur indiquera, selon une des règles de cooccurrence, l'importance de la phrase en question.
5. ***Étude statistique*** : Cette méthode qui peut être considérée comme un processus d'entraînement met en relation un corpus de texte source et leur résumé. Lors de cette étude, des traits et des attributs jugés comme importants sont dégagés. Ces traits peuvent être par exemple :
    a. Simples marqueurs, « *On introduit* », « *en conséquence* »…
    b. La position initiale : apparaît dans l'introduction, la conclusion ou la première phrase d'un paragraphe…
    c. Le contenu d'une phrase, contient des mots du titre ?

A chaque phrase est ainsi associée une probabilité en fonction des listes de traits envisagés. L'interprétation d'un nouveau texte consiste à calculer la probabilité de chaque phrase selon les marqueurs observés. Les phrases qui possèdent la plus grande probabilité sont choisies pour le résumé. Plusieurs combinaisons ou variations de ces méthodes d'extraction sont souvent utilisées dans les résumeurs complexe pour améliorer la précision de la sélection. Cependant, un équilibre entre la complexité, l'espace, le temps d'exécution et la précision limite la fusion de ces méthodes. Dans les systèmes les plus avancés, une étude sémantique basée sur des connaissances du monde (Scenario, scripts…) peut être utiliser pour ajouter à la précision des résumés, cependant cette étude entraîne une dépendance à un domaine contradictoire à notre but original. Dans ce qui suit nous présenterons l'analyse sélective, la méthode de production de résumé utilisée dans ***SumUm***.

2.2 L'analyse sélective

L'analyse sélective est un processus pour la production automatique de résumés spécifiques à un domaine technique « *structuré* » (*Saggion, Lapalme, 2000a*). L'analyse sélective produit un résumé indicatif avec la possibilité d'extension de sujets informatifs selon le choix de l'utilisateur. L'architecture de l'analyse sélective est présentée à la **figure 1**. C'est un processus contenant deux étapes principales :

- L'interprétation qui interprète les phrase lexicalement, syntaxiquement et sémantiquement
- La sélection qui s'occupe de la sélection des phrases pertinentes.

La première étape, PRE-PROCESSING et INTERPRÉTATION, est l'extraction d'information du document source en utilisant une ou plusieurs des méthodes mentionnées dans la section précédente. L'interprétation débute par la segmentation du texte en ses structures, puis par l'application d'un « ***text tagger*** » à chaque structure identifiant pour chaque mot sa catégorie lexicale et sa forme canonique, ces phrases sont ensuite introduites à des automates représentant une grammaire pour produire la représentation syntaxique de phrases. Un dictionnaire conceptuel (CONCEPTUAL DICTIONNARY) est utilisé, dans une interprétation sémantique, pour une première fois dans cette phase pour associer à chaque phrase un concept et une relation. Le dictionnaire conceptuel sera détaillé dans les sections suivantes puisqu'il englobe les patrons, sauvegardés sous forme de structures spéciales appelées template (qui sont vides à ce moment), qu'on essaie d'automatiser. On note pour l'instant que ce dictionnaire est le résultat d'un appariement manuel des documents du corpus et leurs résumés correspondant produits par des résumeurs professionnels. A chaque phrase sont associés une sémantique ainsi qu'un groupe conceptuel selon ce dictionnaire conceptuel. La phase d'interprétation produit une liste des termes du document avec leur fréquence (TERM TREE), une liste des termes qui apparaissent dans les titres des sections (TOPICAL STRUCTURE) et la représentation du texte interprété au niveau lexical, syntaxique et sémantique (TEXT REPRESENTATION).

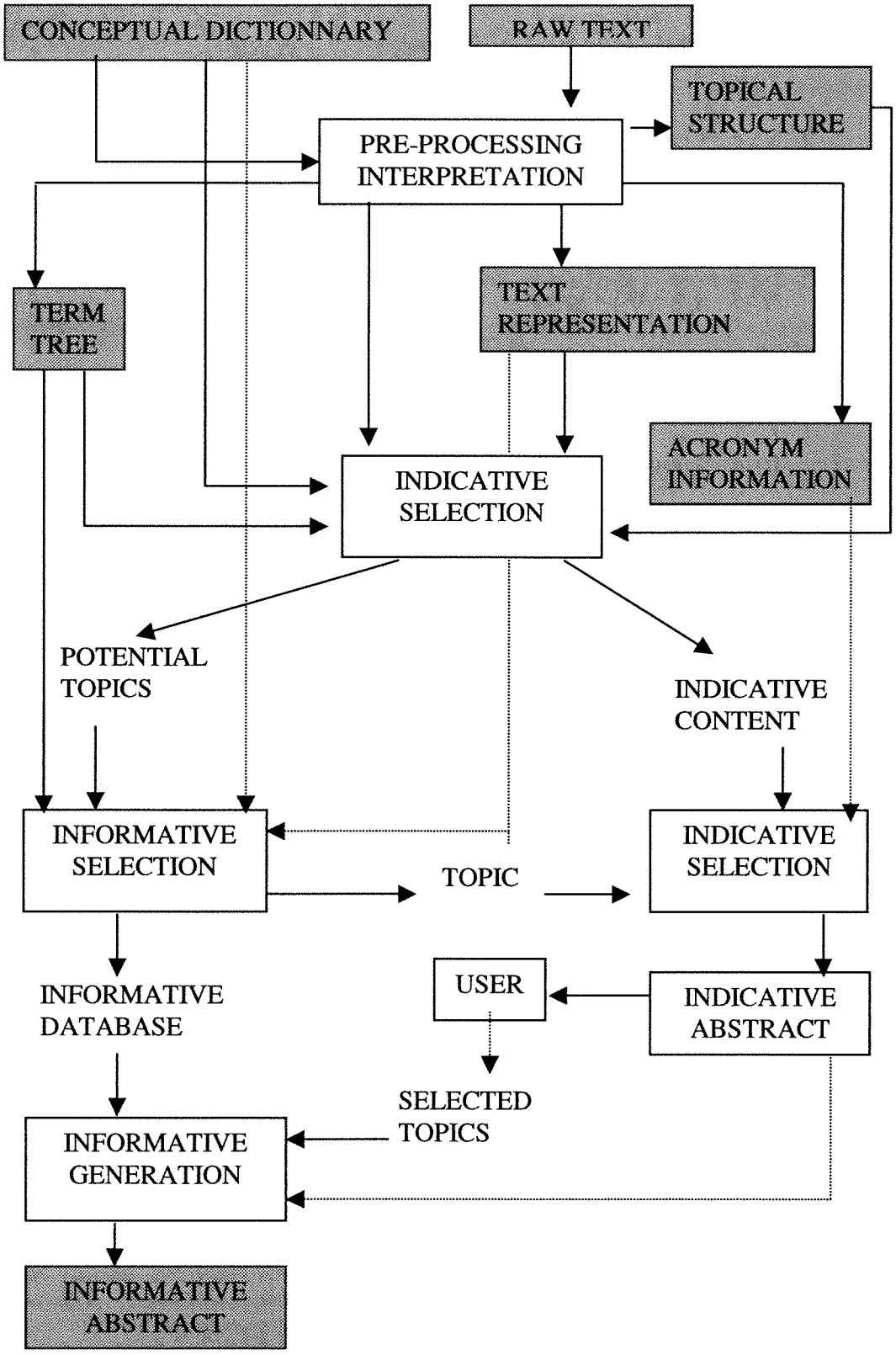


Figure 1: L'analyse sélective

Pendant la phase suivante de la sélection indicative, le dictionnaire conceptuel est consulté dans une seconde passe pour appliquer les patrons à nos phrases déjà interprétées. Dans cette phase chaque phrase est appliquée à un patron, ou plus simplement « le template du patron » (table qui contient les informations sur un patron) est rempli (La sélection est souvent appelée « le remplissage de template »). Des poids sont attribués à chaque phrase selon plusieurs critères (La fréquence, le type de concept, le type de relation, la position) et les « templates » remplis de phrases vont constituer le contenu indicatif (INDICATIVE CONTENT). Le contenu indicatif va être ensuite manipulé et comparé à la structure topique (structure contenant les mots des titres, conclusion et sections du documents) pour choisir les sujets potentiels et le résumé indicatif. On ne va pas détailler cette phase mais plus d'information sur l'association de poids à chaque template et le choix des templates finaux se trouve dans (*Saggion, 2000a*). Les notions de patrons, dictionnaire conceptuel et de template seront explicitées dans les sections suivantes.

Nous avons brièvement présenté la sélection indicative, une méthode de production de résumé de documents « *techniques* ». Nous détaillons maintenant l'utilisation de la sélection indicative dans ***SumUm***, le langage utilisé, le processus d'interprétation et surtout la formation du dictionnaire conceptuel incluant les patrons qui ont été déterminés par un appariement manuel et coûteux du corpus. On présentera un exemple de ***SumUm*** et on parlera de ses avantages et ses limites qui justifieront notre but d'intégration d'un algorithme de découverte de patrons.

2.3 SumUM

SumUM est l'implantation de l'analyse sélective pour la production de résumés automatiques sous les deux formes, indicative et informative. ***SumUM*** est écrit en ***SICStus Prolog*** (*SICStus, 1998*) (*Ross, 1989*) (*Walker, 1987*) sur les systèmes d'exploitation ***Sun 5.6*** et ***Linux*** (RH 6.0). ***SumUm*** utilise deux programmes externes, le premier pour segmenter le texte en section, titre, auteur et références et le deuxième pour effectuer le tagging. On va présenter dans les sections suivantes la structure de ***SumUm,*** la formation du dictionnaire conceptuel et les études faites pour arriver aux patrons. On finira ce chapitre on parlant des limitations de ***SumUm*** et comment l'introduction d'un algorithme d'apprentissage aidera à diminuer ces limites. ***SumUm*** est formé de trois phases : L'interprétation (Lexical, syntaxique et sémantique), la sélection (Application de patrons, Remplissage de template et sélection de template) et la génération. Dans ce qui suit on ne va pas détailler la sélection de template et la génération; Cependant, on va débuter par l'introduction de la structure générale de ***SumUm***, on va ensuite parler du dictionnaire conceptuel et les patrons qu'il contient et finalement on va donner un exemple pour clarifier le tout et situer notre problème.

2.3.1 L'interprétation dans SumUm

La structure de ***SumUm*** est présentée à la **figure 2**. Le texte est d'abord introduit à un segmenteur (SEGMENTATION) qui structure le texte en plusieurs sous sections (AUTHOR, TITLE, REFERENCES et SECTIONS). Chaque sous-section est ensuite introduite à un tagger (POS-TAGGING) qui s'occupe d'associer des catégories lexicales à chaque termes. Les résultats sont des fichiers de format tag (TAGGED TITLE, TAGGED SECTIONS…) contenant les phrases du document avec leurs interprétations lexicales. Le tagger contient en total 215 catégories lexicales, un exemple des différentes catégories lexicales du tagger est présenté dans la **table 1**. Avec le tagging, la première phase de l'interprétation lexicale est accomplie.

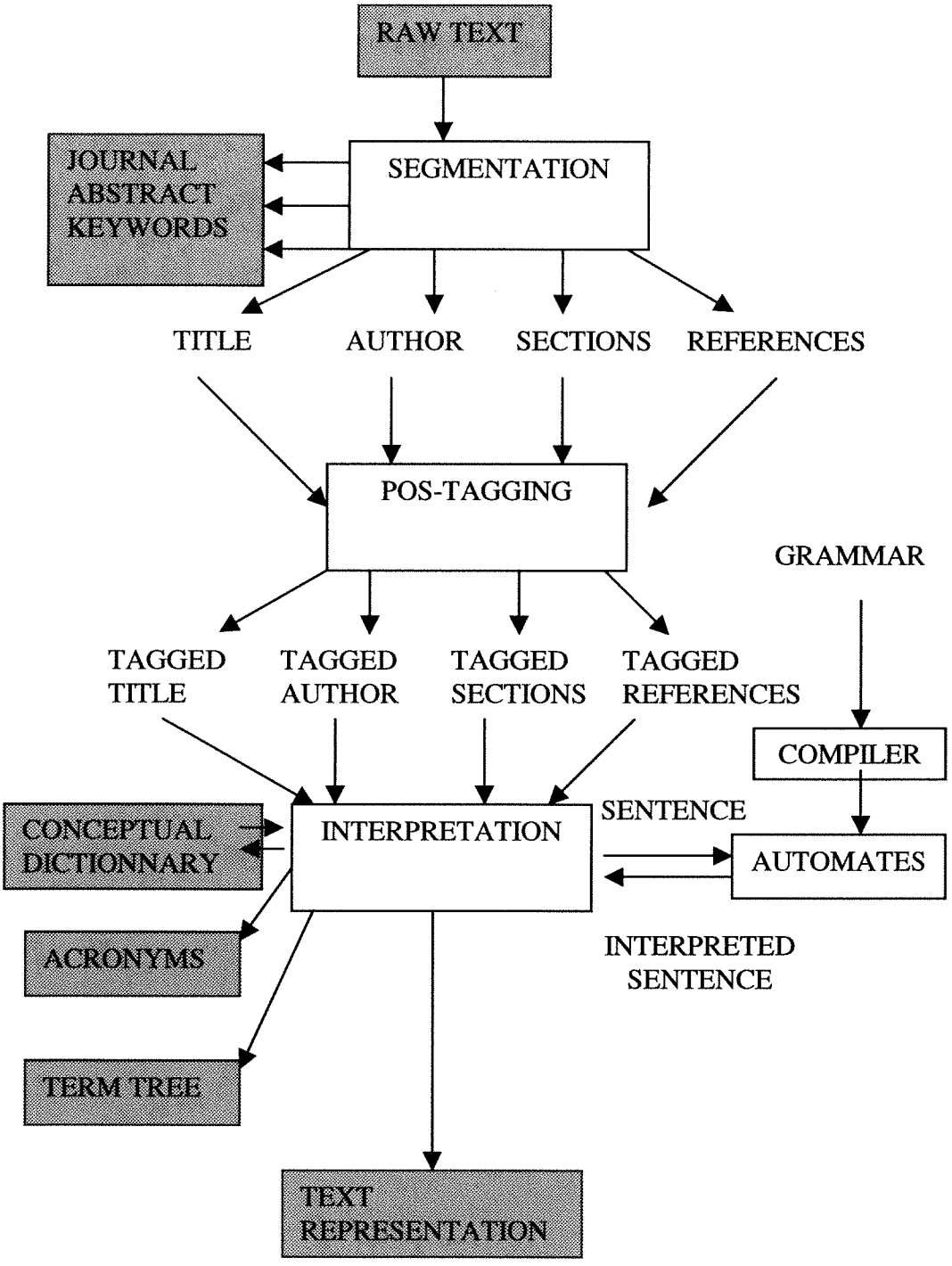


Figure 2: L'interprétation dans SumUm

POS	Meaning	Example
Dete	Determinant	The
Quan	Quantifier	Last
AdjQ	Adjective	substantial
Verb	Verb	Made
PNoun	Proper Noun	CBR
Prep	Preposition	Within
Pron	Pronoun	We
Punc	Punctuation	;

Table 1: Catégories Lexicales

La deuxième phase s'occupe de l'interprétation syntaxique. Les fichiers taggés sont introduits à des automates compilés qui représentent notre grammaire (AUTOMATES) et qui attribuent à chaque phrase le groupe syntaxique convenable : La grammaire est présentée sous forme de FST (Finite State Transducers), chaque FST représente une syntaxe (Groupes nominaux, groupes verbaux, groupe adjective) comme Dete A+ N+, Dete N+, etc.... La phrase interprétée lexicalement est introduite à ces FST récursivement et chaque fois qu'une partie de la phrase est satisfaite par un FST on essaie de trouver la bonne FST pour la reste de la phrase et en sortie la syntaxe de la phrase pour achever ainsi l'interprétation syntaxique. **Table 2** présente un exemple d'une phrase après l'interprétation lexicale et syntaxique.

The computer science dean (Dete A+ N+, Determinant Adjective group Noun Group)
(syncat, gn), (gntype, GN4=[dete, A+, N+]), (string, [A, computer, science, dean]), (canon, [computer, science, dean]), (DeteType, dart), (type, def), (Nbr, sing)...

Table 2: Un fragment d'une phrase interprétée au niveau syntaxique et lexical

La troisième étape reçoit les phrases résultant de l'interprétation lexicale et syntaxique et les combine avec le dictionnaire conceptuel (CONCEPTUAL DICTIONNARY) contenant, à part des patrons, des informations sémantiques sous forme de concept et relation, pour associer à chaque phrase une catégorie sémantique ou simplement une liste de concept et relation et terminer ainsi l'interprétation sémantique (On note que jusqu'à l'instant, les patrons ne sont pas encore utilisés). Les patrons du dictionnaire conceptuel et les résultats de l'interprétation incluant le texte interprété (TEXT REPRESENTATION), l'arbre de terme contenant les fréquences des termes (TERM TREE) et la structure d'expansion des acronymes (ACRONYMS) seront ensuite utilisés dans la phase prochaine de sélection pour le remplissage de structures spéciales appelées templates.

On passe immédiatement à la présentation du dictionnaire conceptuel avec les patrons et les templates qui représentent ces patrons.

2.3.2 Le dictionnaire conceptuel et les patrons

Dans les premières étapes de développement de ***SumUm***, une étude poussée d'un corpus de 120 documents techniques et leurs résumés fournis par des experts humains, a été accomplie dans le but de dégager certaines régularités, certains styles ou plus simplement patrons qui gouvernent ces documents. Le corpus utilisé, qui va être consulté et utilisé plus tard dans l'entraînement de notre algorithme d'apprentissage, est formé de 120 documents. Chaque document est composé du résumé professionnel et du document source. Les sources des résumés utilisés proviennent des journaux « *Library and Information Science Abstracts* » (LISA), « *Information Science Abstracts* » (ISA) et « *Computer Abstracts* ».

Les documents sources se trouvent dans les références suivantes :

- ***« Journals of Computer Science » (CS)***
- ***« Information Science » (IS)***
- ***AI Communications***
- ***AI Magazine***
- ***American Libraries***
- ***Annals of Library Science & Documentation***
- ***Artificial Intelligence***
- ***American Libraries***
- ***IEEE Expert***

Les documents sources sont des documents techniques qui couvrent divers sujets dans CS (Computer Science) et IS (Information Science). Plus spécifiquement 62 documents de CS et 38 de IS. La majorité des documents sont structurés (Introduction, Conclusion et parfois section, référence, auteur…) et sont de 7 pages en moyenne variant entre 2 et 45 pages.

L'analyse préliminaire du corpus de documents et de leurs résumés a souligné l'importance de plusieurs critères qui seront utilisés plus tard dans notre algorithme d'apprentissage :

1. ***La position de la phrase :*** la majorité des informations critiques viennent de l'introduction, la conclusion ou bien les titres. En fait, 72% des informations pertinentes proviennent de ces parties (**Table 3**).
2. ***L'occurrence de certains concepts et relations*** sous des patrons spéciaux.
3. Voix et conjugaison des groupes verbaux.

	Documents	
	#	**%**
Title	10	2
Author Abstract	83	15
First Section	195	34
Last Section	18	3
Subtitles & Capts.	191	33
Other Sections	71	13

Table 3: Distribution de l'information

En plus, environ 205 phrases alignées des documents sources contiennent des termes lexiques spécifiques au domaine technique, soit 35% des phrases alignées. 35% de phrases proviennent aussi des titres, introduction et conclusion, pour un total de plus de 70% des phrases alignées indiquant les « *expressions indicatives* » contenant un patron régulier.

Suite à cette étude de corpus, les observations obtenues ont été organisées dans une base de donnée appelée le dictionnaire conceptuel. Ce dictionnaire inclut trois listes :

- Une liste de concepts et une liste de relations représentant la sémantique. Cette liste est utilisée par ***SumUm*** pendant la phase de l'interprétation pour associer des concepts et relations à chaque phrase.
- Une liste des patrons réguliers gouvernant ces concepts et relations sauvegardée sous des structures appelées « template ». Ceux sont les patrons que notre algorithme va automatiser.

Les concepts :

Les concepts (**table 4**) englobent les connaissances et les sémantiques des groupes nominaux en général. Le concept « *author* » décrit les auteurs de l'article et peut être indiqué par les mots « *we, I, author...* ». Cinquante-cinq concepts ont été déterminés. Une liste complète se trouve à **l'annexe A**.

Concepts	Explication et Exemple	Mots lexicaux
Research	The research work. "...some scientific research..."	Research,...
Institutions	Institutions. "Department of computer science.."	University, department...
Project	A research project. "A project currently in progress..."	Project,...

Table 4: Liste de quelques concepts

Les relations :

Les relations présentent les sémantiques des groupes verbaux. La relation « *make known* » par exemple, introduit le sujet principal du document. Cette relation est produite par différents termes lexicaux, « *describe, expose,...* » : « *In this paper we present...* » englobe la relation « *make known* ». Voici quelques exemples de relations (**table 5**), la liste complète se trouve a **l'annexe B** :

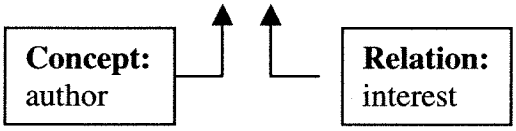
Relations	Explication et Exemple	Mots lexicaux
Show	Identifying graphical material. "Figure 1 illustrates the concept..."	See, show, ...
Make known	Introducing the topic. "In this paper we present..."	Describe, expose, ...
Elaborate	Elaborating. "This property allows us to us..."	Allow, contribute,...

Table 5: Liste de relations

Les patrons ou types d'information:

Les patrons ont été formés par comparaison de phrases, observation de cooccurrence de concepts et relations, des positions, des voix et d'autres critères. Voici quelques exemples :

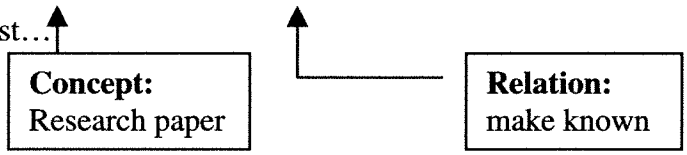
Author Interest : Identifié par la cooccurrence du concept « *author* », et la relation « ***interest*** » : « Finally *we address* the issue of scalability.... »



Le patron est présenté sous la forme suivante *: Skip1+Author+Interest+XX+eos. Skip1* est une variable à rejeter qui sert à sauter jusqu'au concept *Author. XX* est une variable qui contient le fragment qui suit la relation *Interest.* Dans ce cas *Skip1* est instancié à « *Finally* » et *XX* à « *the issue of scalability...* ». *eos* porte le point indiquant la fin de la phrase.

Topic of document : L'auteur marque le sujet du document. Cela est identifié par la cooccurrence de la relation « *make known* » et un des concepts « *author* » ou « *research paper* » dans la première ou dernière section du document.

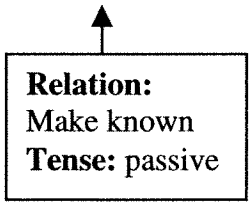
Ex. : In *this paper we have presented* a more efficient algorithm to construct breadth first...



Le patron est: Skip1+Research Paper+Make known+XX+eos.

Possible Topic: Identifié par la présence d'un verbe de la relation "*make known*" dans une forme passive dans la première ou dernière section du document.

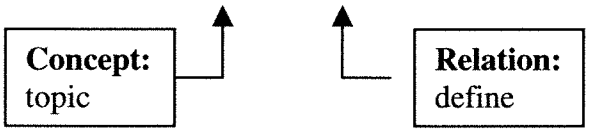
Ex. : The sampling procedure is *described*, in which queries obtained....



Le patron est: XX+Make known+Skip1+eos.

Topic: Un sujet est introduit explicitement par le concept « *topic* » ou « *conceptual topic* » et la relation « *define* ».

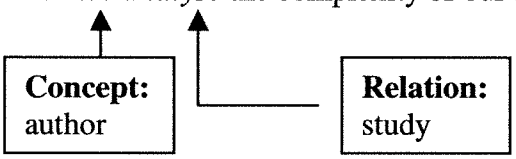
Ex. : *The subject of this paper is* the concept of descriptor equivalence…



Le patron est : Skip1+Topic+Define+XX+eos.

Author study: Introduction à une étude d'un concept. Cooccurrence du concept « *author* » et la relation « *study* ».

Ex. : *We analyse* the complexity of our algorithm



Le patron est : Skip1+Author+Study+XX+eos.

Ces patrons sont sauvegardés dans le dictionnaire conceptuel dans des structures sous forme de tables appelées « templates ». Chaque patron est ainsi associé un template qui débute vide et est remplie graduellement durant la phase de la sélection. On présente ci-dessous la sélection indicative. Une présentation du dictionnaire conceptuel se trouve à la **figure 3**.

Liste de concepts (référence complète à l'annexe A)	***Liste de relation (référence complète à l'annexe B)***	***Liste de templatse vides représentant chaque patron***

Figure 3: Le dictionnaire conceptuel

2.3.3 Sélection indicative, un remplissage de template

La sélection indicative comporte principalement un remplissage des templates par consultation du dictionnaire conceptuel. Dans cette phase, chaque phrase

interprétée est associée à un ou plusieurs templates. Les templates sont plus ou moins des tables structurées, des objets contenant des attributs, appelés slot ou filler. Un exemple du template représentant le patron défini par la cooccurrence de la relation "*make known*" et le concept « *author* » dans la première ou dernière section du document ou tout simplement le patron appelé « Topic of Document» est présenté à la **table 6.**

Topic of Document	
Type	Topic of Document
Id	3
Pattern	Skip1+Author+Make Known+XX+eos.
Position	Section 1, Last Section
Topic Candidates	
Weight	

Table 6: Le template Possible Topic

Chaque template est associé un « *Id* », un nombre qui identifie uniquement chaque template. Le patron (*Pattern*) contient la définition du patron : Dans notre exemple, *Skip1+Author+Make Known+XX+eos*, identifie un patron de cooccurrence du concept « *Author* » et de la relation « *Make Known* ». « *eos* » est utilisé pour délimiter et indiquer la fin de la phrase, « *Skip1* » est une variable qui contient les terme qui apparaissent avant le concept « *Author* ». « *XX* » est la variable qui contient les sujets possibles « *Topic Candidates* » qui seront sauvegardés dans le « slot » « *Topic Candidates* ». Considérons la phrase suivante :

> In the course of this study, *we have presented* a more efficient method to test soil productivity.

SumUm débute par l'interprétation de chaque phrase du document, en particulier la phrase précédente, le tagging associe les catégories lexicales, les automates donnent les catégories syntaxiques et le dictionnaire conceptuel associe les concepts et relations convenables à cette phrase (partie sémantique). On note que le

dictionnaire conceptuel contient aussi la liste de patron sous forme de templates vides qui ont été produis par l'étude de corpus.

Dans la seconde étape de la sélection indicative et remplissage de patrons, cette phrase sera associée au template de la **table 6** puisqu'elle satisfait le patron et la position de ce template. « *In the course of this study* » sera unifié avec la variable « *Skip1* », « *we* » avec le concept « *Author* », « *Have Presented* » avec la relation « *Make Known* » et « *XX* » avec « *a more efficient method to test soil productivity* ».

Le slot « *Topic Candidates* » sera rempli avec la valeur de « *XX* ». Le « *Weight* » à son tour sera calculé pour chaque phrase associée à un template selon des critères qu'on ne va pas détailler comme la fréquence de termes, le type de concept ou de relation, la position... (*Saggion, 2000d* contient plus de détail sur ce sujet).

Quelques exemples de patrons sont présentés à la **table 7**.

Type	Spécification de patron
Signaling	*SKIP1+structural+SKIP2+show graphically+XX+eos*
Topic	*Gn+author+make known+Prep+research paper+XX+eos*
Definition of TOPIC	*SKIP+XX+define+gn*
Elaboration of TOPIC	*SKIP+XX+to elaborate*

Table 7: Patron des types d'information

Les templates ainsi remplis sont sauvegardés dans une liste spéciale appelée la base de donnée indicative ou IDB. Cette base est appariée à la structure topique (Contenant les termes du titre, conclusion et titres des sections) pour sélectionner le contenu indicatif (Sélection de contenu) et les sujets potentiels (en réalité la sélection de contenu est plus complexe mais on omis ces détails puisque notre intérêt est porte sur la formation et le remplissage de template non pas sur la sélection des template finaux). Le processus de sélection et remplissage de template est décrit à la **figure 4.**

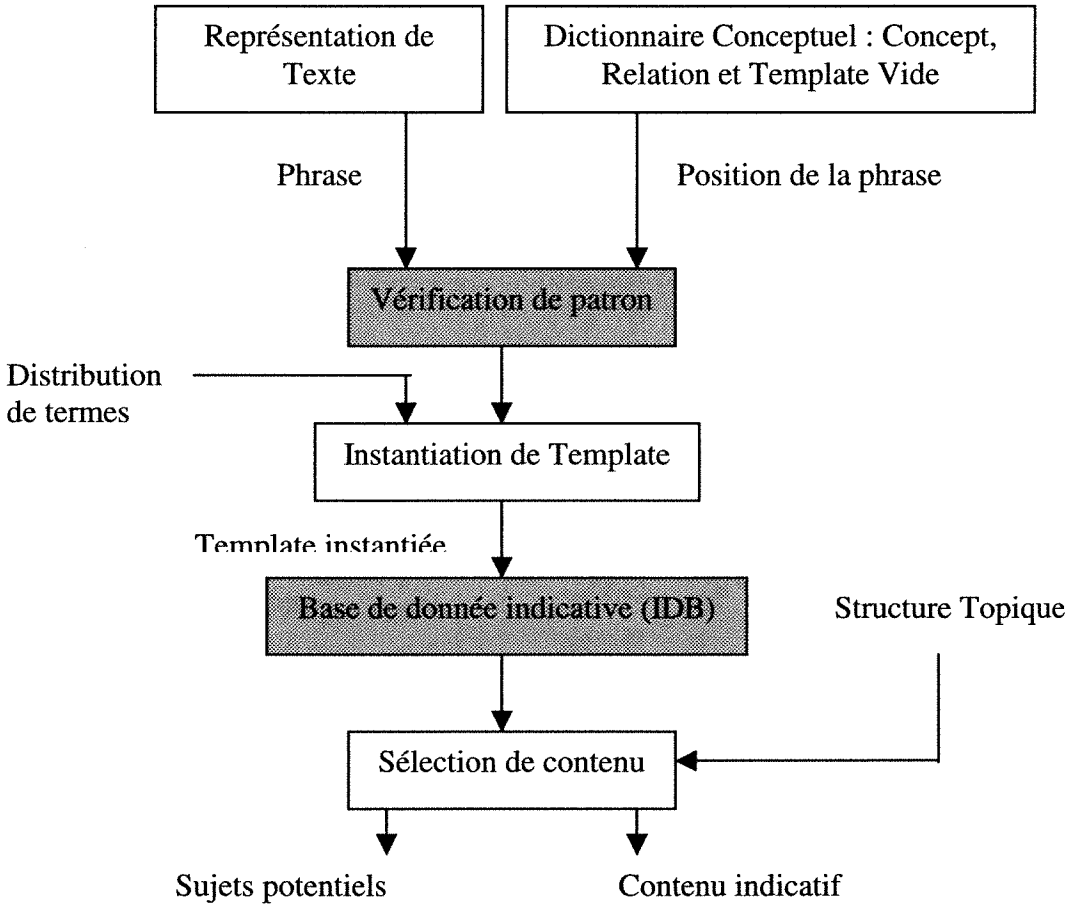


Figure 4: Sélection et Remplissage de Template

Après la sélection du contenu, une dernière phase de génération se fait par la transformation des templates choisis avec quelques modifications :

- *Ré-formulation de verbe :* Changement de voix (active/passive), position (début, centre ou fin de la phrase), temps et le nombre.
- *Ré-formulation de concept :* transformation simple de pronom, préposition et déterminant.
- *Expansion des acronymes :* Consultation des formes canoniques

Les phrases du résumé final seront différentes d'une imitation exacte du texte source amenant ainsi un certain degré d'originalité. Un exemple d'une sortie typique de ***SumUm*** est présenté à la **figure 5**. « *WC* » représente le « *Word Count* » ou nombre de mots qui apparaissent dans l'abstract alors que les « [ ] » représentent les sujets identifiés et peuvent être élaborés selon la demande de l'usager.

> ***Presenting the indicative abstract...***
> The complex problem of inspection and maintenance of the steam generator in nuclear power plants was approached using previously gained expertise and, as a result, an innovative solution was achieved with the development of two co-operative robots, remotely controlled from a tele-operation station incorporating tele-presence. The department was leading a project for the introduction of a robotics system whose mission was to avoid human operators having to enter the steam generator's water chamber. Proposes an innovative solution to the serious problem of inspection and maintenance of the steam generator tubes, which must be checked regularly to detect leakage of radioactive water from the primary to the secondary circuit. Another interesting activity was the realization, within the framework of a EUREKA project. Of a tele-manipulator for servicing a new concept of urban infrastructures. Shows the RIMHO walking robot and ROBUR arm exchanging gas filter in IUI (Industrializable urban infrastructures) demonstration.
>
> ***Abstract WC***: 149
> ***Identified Topics***: [IUI] [RIHMO] [chamber] [climb, robot] [control, station] [control, system] [different, robotic, system] [human, operator] [industrial, robot] [infrastructure] [robot] [robot, arm] [system] [water, chamber] [whell, mobile, robot] [whole, control, system]

Figure 5: Un résumé de SumUm

On a présenté ***SumUm***, un prototype d'Analyse Sélective qui transforme un texte source en un résumé. La comparaison de ***SumUm*** avec divers systèmes de productions de résumés (**Microsoft'97 Summarizer**, **Word Distribution**, **Extractor**, **n-STEIN**) sont plus que satisfaisante, en effet ***SumUm*** est, la plupart du temps, plus efficace. Cependant à quel prix obtient-on cette pertinence?

2.4 Limitations de SumUM

Lors du développement d'un système de production de résumés automatiques, on doit choisir entre le général et le spécifique, entre la puissance et le coût d'analyse. ***SumUM*** est plutôt axé sur le spécifique et la puissance d'analyse. Comme tout autre système de production de résumé automatique complexe, ***SumUm*** souffre d'une dépendance à un domaine spécifique causée par deux points :

1. Il s'appuie sur la structure du document, le titre, la conclusion, les références et les sections. Il utilise une liste des termes et des fréquences qui apparaissent dans les phrases de la structure principale du texte, la structure topique - comme une partie essentielle de la sélection indicative des phrases pertinentes (puisque à la fin, ***SumUm*** compare les templates remplis à la structure topique pour dégager les templates qui forment le résumé final).
2. Il utilise le dictionnaire conceptuel, contenant les patrons (sauvegardé dans des templates) dégagés par appariement de corpus de documents techniques et qui sont spécifiques à un domaine.

On peut envisager que le premier point de la structure du document peut être résolu en substituant la segmentation par section en une segmentation par paragraphe ou autre technique tout en tenant compte du bruit que la méthode choisie entraîne. Dans ce cas, le vrai problème qui reste est celui du choix des patrons, qui sont une partie intégrale de la sélection indicative et de ***SumUm,*** et qui dépendent du domaine. Après tout, bien que la cooccurrence du concept « *author* » et de la relation « *make known* » soit considérée comme un patron dans les documents techniques, elle ne l'est pas dans d'autres types de documents. On peut parfaitement conclure que les patrons soulèvent le plus grand problème de limitation qui empêche ***SumUm*** d'être généralisé pour englober tous les domaines et ne pas se restreindre à la production des résumés techniques.

2.4.1 Le choix de patrons

Le choix de patrons a été fait par une étude poussée d'un corpus de documents techniques avec leur résumé produit par des résumeurs professionnels. Les phrases du corpus des documents sources ont été alignées avec les phrases correspondantes des résumés, les similarités, les cooccurrences de concepts, relations et positions ont été notées pendant ce processus qui en plus de sa complexité, exige un travail à la main très fastidieux. Des patrons ont ainsi été formés selon ces critères et sauvegardés dans les templates. Ces patrons sont dépendants du domaine, « the author present », « we described », « the writer introduced » sont des patrons (concept « author », relation « make known ») fréquents dans les documents techniques mais inexistants dans les documents qui parlent des de meurtres dans un journal où des patrons comme « Suspect shoots », « Ex-con kidnaps » sont beaucoup plus communs.

Notre but est d'automatiser la détermination de patrons faite par l'étude de corpus Ces patrons qui malgré qu'ils soient valables pour les documents techniques, ne le sont pas pour les documents d'un autre domaine. On espère arriver à un système qui peut changer dynamiquement ses patrons par ré-entraînement selon le domaine désiré. Le scénario optimal est le suivant : Pour faire un résumé des documents d'un journal, on entraîne le système par un corpus de documents de journal pour former les patrons spécifiques à ce domaine et qui seront sauvegardés dans des templates dans le dictionnaire conceptuel. Notre système sera ainsi prêt à produire un résumé de ce domaine. Si on désire produire un résumé pour un autre domaine, un ré-entraînement, une re découverte de patron et une sauvegarde de ces patrons dans les templates du dictionnaire conceptuel est nécessaire.

2.5 Besoin d'apprentissage

La majorité des domaines d'étude de la langue naturelle reposent sur une extraction de patrons : les programmes de recherche d'information, les systèmes de traduction, les algorithmes de reconnaissance de voix (*Moore, 1994*) ou même les systèmes de réponse automatique aux messages utilisent une extraction et identification de patrons. Cette étude, est cependant, toujours non-automatisée, elle est faite à la main, le plus souvent par alignement et comparaison de corpus. C'est une tâche coûteuse occupant beaucoup de temps et d'effort qu'on va essayer de résoudre. Avec l'introduction de l'apprentissage dans le traitement de la langue naturelle, de nouveaux horizons de recherche ont été développés, en particulier l'apprentissage et l'automatisation de patrons. Les algorithmes de généralisation, des méthodes de spécialisation et d'induction, les arbres de décision, les entraînements et les ré-adaptations, les réseaux neuraux et les systèmes de groupement et de découverte constituent une portion du monde de l'apprentissage qui peut aider au traitement de la langue naturelle, et plus spécifiquement l'apprentissage et le filtrage de patrons (*P. Cohen, V. Chaudhri, A. Pease, B. Schrag, 1999*).

Dans le chapitre suivant, une introduction à l'apprentissage est présentée, suivie de différentes classes de systèmes d'apprentissage. Nous présentons plusieurs algorithmes utilisés en partie pour le développement de notre algorithme de détection des patrons dans ***SumUm*** et qui en facilitera le passage à un autre domaine.

Chapitre 3

3 L'apprentissage informatique

En informatique, l'apprentissage peut être défini comme le ou les changements dans un système le rendant capable d'exécuter une ou plusieurs tâches plus efficacement lors ses prochaines exécutions. Ces changements prennent deux formes principales: Le système peut acquérir de nouvelles connaissances provenant de sources externes ou il peut se modifier en exploitant plus efficacement ses connaissances actuelles.

Les recherches dans le domaine de l'apprentissage peuvent être appliquées à plusieurs domaines : la recherche et la réutilisation de connaissances pour l'amélioration des décisions (p.e diagnostic médical), les applications difficiles à programmer (p.e reconnaissance de la voix) ou des applications qui changent et s'adaptent (*Moore, 1998*). Les expériences changent l'état d'un système de façon à ce qu'il devienne meilleur à l'avenir .

Expérience ⟶ **État** ⟶ **Performance**

Figure 6: Changement de la performance d'un état par l'introduction de nouvelles expériences

Un système améliore ses expériences par observation des exemples et par un entraînement qui aboutit à un changement du système qui, idéalement, améliore sa performance :

1. Une première étape **d'expérience** débute par l'introduction d'un groupe d'entraînement. Dans notre cas, un groupe de documents du corpus est sélectionné comme base d'entraînement. Les phrases sont automatiquement appariées, des expériences sont acquises.
2. Une deuxième étape de changement **d'état** pendant laquelle le système tire avantage des informations dégagées par les expériences de la première étape pour se modifier. Dans notre cas, un remplissage des patrons

découverts et une intégration de ces patrons dans des templates du dictionnaire conceptuel de ***SumUm*** constitue le changement d'état.

3. Une troisième étape d'évaluation de **performance**, le système modifié (nouvel état résultat de la deuxième étape) est testé pour consulter sa performance. Dans le cas d'une détérioration de la performance, un nouveau groupe d'entraînement est introduit ou tout simplement l'apprentissage est jugé comme échec. Dans notre cas, notre nouveau système ***GlobSum***, va être testé dans deux évaluations pour noter la différence de performance et la qualité des résumés.

3.1 L'histoire de l'apprentissage

L'histoire de l'apprentissage peut être divisée en trois périodes:

- L'exploration (1950... 1960).
- Développement d'algorithmes (1970… 1980).
- Explosion des directions de recherche (1980 et plus).

Les premiers travaux des années 1950 et 1960 étaient inspirés par les recherches biologiques, neurophysiologiques et psychologiques. Plusieurs systèmes ont été développés, comme le perceptron de Rosenblatt's (*Rosenblatt 1958*), les systèmes de sélection naturelle ou les systèmes de représentation symbolique. Les analyses théoriques de l'apprentissage de machine ont débuté dans les années 1960, en particulier les travaux cognitives de Reitman (*Reitman 1965*) et le GPS de Ernst (*Ernst, 1969*). L'apprentissage est devenu beaucoup plus actif avec la publication des articles sur le monde des blocs, qui a été suivi par plusieurs démonstrations de l'apprentissage comme les systèmes METADENDRAL, AQ11 et ID3. Dès les années 1980, l'intérêt dans les travaux d'apprentissage a explosé. Au moins huit classes de recherches sont formées : *Théorie d'apprentissage, Algorithmes d'apprentissage symbolique, Algorithmes de découverte et de groupage, Système d'explication, Apprentissage Inductif, Raisonnement Analogique, Algorithmes Génétiques.*

Ces différentes classes d'apprentissage avec leurs caractéristiques correspondantes sont présentées dans la section suivante.

3.2 Les différentes classes d'apprentissage

Le premier type d'apprentissage, obtenu à l'aide d'un raisonnement basé sur des exemples externes pour produire des règles générales, est appelé apprentissage inductif ou empirique. Le théorème de l'apprentissage inductif dit qu'une hypothèse capable de prédire la fonction but sur un nombre suffisamment grand d'exemples d'entraînement, est aussi capable de prédire cette fonction sur de nouveaux exemples non observés.

Une grande partie de l'apprentissage inductif prend la forme des algorithmes SBL (*similarity-based learning*) qui fonctionnent par la comparaison entre les exemples d'entraînement pour trouver les similitudes. L'apprentissage inductif, à son tour, est divisé en deux types (*Shavlik, 1990*):

1. L'apprentissage supervisé.
2. L'apprentissage non-supervisé.

L'apprentissage supervisé est un type d'apprentissage inductif où le programme reçoit des exemples sous la forme $(\mathbf{x_i}, \mathbf{y_i})$ pour apprendre une fonction **f** tel que $\mathbf{f(x_i)=y_i}$ pour tout **i**. En plus, cette fonction **f** doit capturer la configuration générale dans la structure d'entraînement, ainsi **f** peut être appliquée pour prédire les valeurs de **y** pour les nouvelles valeurs de **x**. Typiquement, chaque **x** est la composition de descriptions d'un objet ou situation, alors que chaque **y** est une description plus simple. Les valeurs de $\mathbf{y_i}$ sont fournies par un instructeur/superviseur, d'où le nom "apprentissage supervisé". Quand il n'y a que quelques valeurs de $\mathbf{y_i}$, on les appelle classes, et la fonction **f** assigne chaque $\mathbf{x_i}$ à une classe. Dans le cas particulier où seulement deux valeurs de $\mathbf{y_i}$ sont disponibles, les exemples d'entraînement sont classés comme positifs et négatifs. **f** est alors dit une fonction de définition de concept, cette tâche est l'apprentissage conceptuel.

L'apprentissage non-supervisé est composé de deux types: Groupage et Découverte. Le programme d'apprentissage reçoit une collection de valeurs $\mathbf{x_i}$ pour en chercher des régularités, qui diffèrent d'une application à l'autre. La majorité de ces régularités se trouvent sous la forme de groupes de certaines valeurs de $\mathbf{x_i}$, c'est l'apprentissage par groupage (*clustering*) où une valeur de similarité ou dissimilarité est toujours requise pour guider l'algorithme dans ces décisions (*Everitt, 1980*).

D'autre part, les programmes de découverte cherchent des relations plus complexes entre les valeurs de $\mathbf{x_i}$. Le second type d'apprentissage, où le système améliore sa performance en exploitant ses connaissances actuelles plus efficacement, est appelé apprentissage accéléré. Il dépend souvent de méthodes d'amélioration de l'espace de recherche. Une de ces méthodes est l'introduction des opérateurs macro capables d'effectuer des sauts dans l'espace de recherche. Ces opérateurs peuvent réduire la profondeur de la recherche nécessaire pour la transition entre l'état initial et l'état final, mais ils peuvent aussi augmenter le facteur de branchement de chaque état.

Dans ce qui suit, nous présentons les deux principaux types d'apprentissage : supervisé et non-supervisé. Bien que notre algorithme soit supervisé, il utilise quand même plusieurs caractéristiques des algorithmes non-supervisés.

Algorithmes Supervisés

Les premiers travaux ont porté sur l'apprentissage supervisé. Ces explorations des différentes techniques ont abouti à la formulation de plusieurs algorithmes. On citera les méthodes les plus connues avec leurs caractéristiques qu'on a utilisées en partie dans le développement de notre algorithme.

Algorithme Find-S:

Cet algorithme, présenté à la **figure 7**, est relativement simple. **H** étant l'espace des hypothèses présentées sous formes d'attributs (attrib1, attrib2,...,attribn) et **x** formé des attributs $\mathbf{a_1....a_n}$.

Initialize ***h*** *to the most specific hypothesis in* ***H*** *(∅, ∅, ..., ∅)*
For *each positive training instance* ***x***,
    If *the constraint* a_i *in* ***h*** *is not satisfied by* ***x*** ***then***
        replace a_i *in* ***h*** *by the next more general*
        constraint that is satisfied by ***x***
Output hypothesis ***h***

Figure 7: L'algorithme Find-S

L'algorithme ***Find-S*** (*Mitchell, 1997*) est considéré comme une des premières méthodes qui a abouti au développement des méthodes d'apprentissage. L'algorithme est simple et se base sur la découverte des hypothèses les moins générales qui satisfont tous les exemples positifs. La similarité de l'algorithme ***GlobSum*** et l'algorithme **Find-S** est dans l'idée de la généralisation par étape que **Find-S** effectue. L'algorithme débute par l'hypothèse la plus spécifique (normalement l'hypothèse vide), et généralise cette hypothèse chaque fois qu'il ne réussit pas à classifier un exemple d'entraînement positif qu'il rencontre, pour couvrir cet exemple rencontré. Le résultat final est une seule hypothèse : La plus spécifique qui couvre tous les exemples positifs. La grande contrainte de l'algorithme ***Find-S*** est la manque de considération des exemples négatifs qui empêchent les généralisations excessives.

Version Space Search:

La recherche dans l'espace de version implante l'apprentissage inductif comme une recherche dans l'espace des concepts. Ces recherches sont basées sur la notion que les opérations de généralisation imposent un ordonnancement des concepts dans l'espace. Cet ordonnancement est utilisé pour guider la recherche. Les généralisations et spécifications sont les types les plus utilisés et communs pour la définition de l'espace de concepts, remplacement de constantes par des variables, élimination de conditions des expressions conjonctives ou ajout à une expression disjonctive (*Mitchell, 1977*). Il existe plusieurs types d'algorithmes pour la recherche dans l'espace de version. Ces algorithmes essaient de réduire le volume

de l'espace de version au fur et à mesure que les nouveaux exemples d'entraînement deviennent disponibles et de créer des partitions de l'espace (*Jan, 1986*). Un de ces algorithmes essaye de réduire l'espace de version dans la direction « *général* → *spécifique* », alors que d'autres adoptent la direction « *spécifique* -> *général* ». Les algorithmes d'élimination de candidats (*candidate elimination*), combinent les deux approches dans une recherche bidirectionnelle.

Ces algorithmes examinent les données et généralisent selon des régularités trouvées dans les exemples d'entraînement (*Mitchell, 1982*). Ces algorithmes produisent donc une variété d'apprentissage supervisé. La recherche dans l'espace de version utilise les exemples positifs et négatifs. Les exemples positifs sont utilisés pour la généralisation alors que les exemples négatifs bloquent les généralisations extrêmes (*overgeneralization*). Le concept à apprendre ne doit pas être seulement suffisamment général pour couvrir les exemples positifs, mais aussi suffisamment spécifique pour exclure les exemples négatifs. La **figure 8** montre un exemple d'un monde de généralisation de l'objet « *ball* » à un objet quelconque « *X* ».

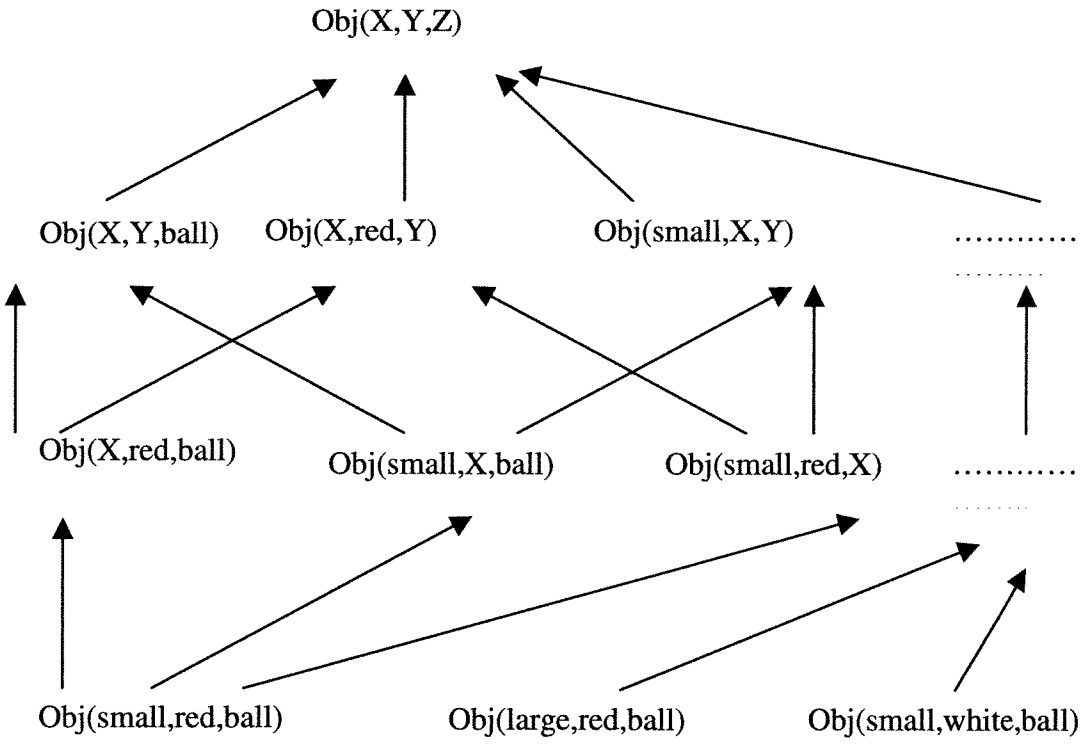


Figure 8: Un espace de concept, du général (plus haut) au spécifique (plus bas)

Dans la **figure 8**, le concept obj(X,Y,Z) couvre tout le groupe de concepts positifs. Ce concept est cependant très général. Une façon d'éliminer la généralisation excessive et d'effectuer une généralisation minimale à chaque étape ou d'utiliser les exemples négatifs pour éliminer les concepts très généraux. Ayant la connaissance qu'un objet petit, rouge et rond est une balle et qu'un objet petit, jaune et rond est aussi une balle, une première généralisation peut conclure qu'une balle est un objet rond et petit.

La recherche dans la direction « *spécifique -> général* » de l'espace de version du concept « *ball* » est présentée dans l'algorithme suivant (**figure 9**), la **figure 10** illustre l'application de cet algorithme sur l'espace de la **figure 8**.

Begin
Initialize ***S*** *to the first positive training instance;*
N *is the set of all negative instances seen so far;*
For *each positive instance* ***p***
    Begin
    For *every* ***s*** *in* ***S, if s*** *does not match* ***p then*** *replace* ***s*** *with its most specific generalizations that match* ***p****;*
    Delete from ***S*** *all hypotheses more general than some other hypothesis in* ***S;***
    Delete from ***S*** *all hypotheses that match a previously observed negative instance in* ***N;***
    End
For *every negative instance* ***n***
    Begin
    Delete all members of ***S*** *that match* ***n****;*
    Add ***n*** *to* ***N*** *to check future hypotheses for overgeneralization;*
    End
End

Figure 9: L'algorithme de recherche de version d'espace du spécifique au général

Cette recherche forme un groupe **S** de concepts candidats. Ces candidats sont les généralisations les plus spécifiques (*maximally specific*) pour éliminer la généralisation extrême. Un concept « *a* » est spécifique d'une manière maximale, s'il couvre tous les exemples positifs, aucun des exemples négatifs, et pour tout autre concept « *b* » qui couvre les exemples positifs, *a* est plus petit que *b*.

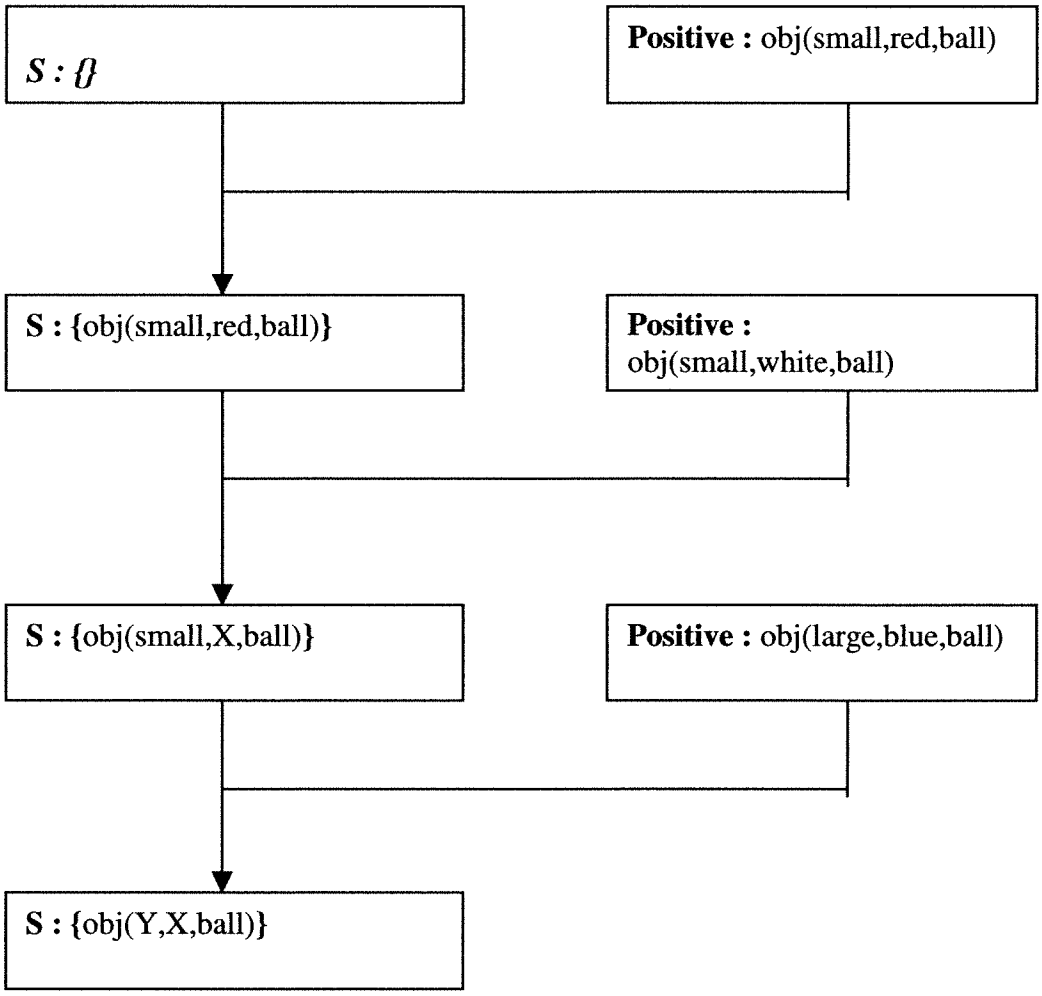


Figure 10: Le résultat de l'application de l'algorithme de la figure 9 sur l'espace de la figure 8

S est graduellement rempli et généralisé au fur et à mesure qu'on introduit de nouveaux exemples : Après l'introduction des deux exemples positifs *obj(small, red, ball)* et *obj(small, white, ball)*, **S** induit la généralisation *obj(small ,X, ball)*. Un

exemple positif additionnel *obj(large, blue, ball)* généralise suffisamment le premier attribut pour y arriver à *obj(Y, X, ball).*

La recherche « *général -> spécifique* » dans l'espace de version pour l'apprentissage du concept « *ball* » est présentée dans l'algorithme de la **figure 11**, l'illustration se trouve dans la **figure 12**:

```
Begin
Initialize G to contain the most general concept in the space;
P contains all positive examples seen so far;
For each negative instance n
        Begin
        For each g in G that matches n, replace g with its most
        general specializations that do not match n;
        Delete from G all hypotheses more specific than some other
        hypothesis in G;
        Delete from G all hypotheses that fail to match some positive
        example in P;
        End
For each positive instance p
        Begin
        Delete from G all hypotheses that fail to match p;
        Add p to P;
        End
End
```

Figure 11: L'algorithme de recherche de version d'espace du général au spécifique

L'algorithme de recherche « *général au spécifique* » forme un groupe **G** de concepts les plus généraux (*maximally general*) qui couvre tous les exemples positifs et aucun des exemples négatifs. Un concept « *a* » est général d'une manière

maximale s'il ne couvre aucun des exemples négatifs, et pour n'importe quel concept « *b* » qui couvre aucun exemple négatif, *a* est plus petit que *b*. Dans cet algorithme les exemples négatifs aboutissent à la spécialisation des concepts candidats; Les exemples positifs sont utilisés pour éliminer les concepts très spécialisés (*overly specialized concepts*).

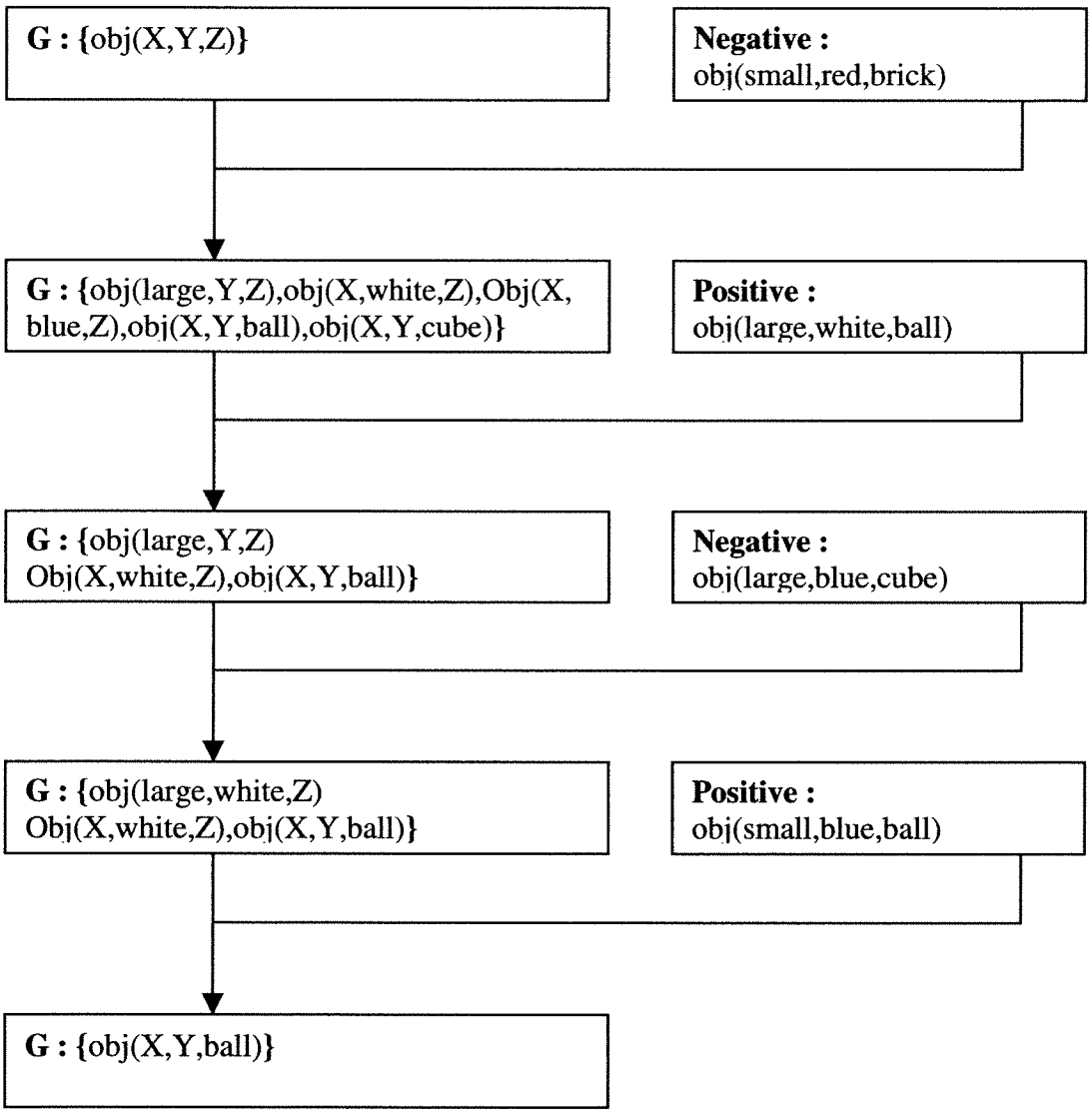


Figure 12: Le résultat de l'application de l'algorithme de la figure 11 sur l'espace de la figure 8

Dans ce cas, **G** est graduellement rempli et spécialisé au fur et à mesure qu'on introduit de nouveaux exemples : Avec l'introduction d'un exemple négatif comme

obj(small, red, brick), on remplace chaque élément de **G** par la spécialisation la plus générale qui exclut l'exemple négatif *obj(small, red, brick)*, avec cette introduction l'ensemble **G** *{obj(X,Y,Z)}* devient alors *{obj(large,Y,Z) ,obj(X,white,Z), obj(X,blue,Z) ,obj(X,Y,ball), obj(X,Y,cube)}*. L'introduction d'un exemple positif exclut à son tour les concepts dans **G** qui ne respectent pas ce concept positif. Après l'introduction du concept positif *obj(large,white,ball)*, **G** est l'ensemble *{obj(large,Y,Z), obj(X,white,Z), obj(X,Y,ball)}*.
La combinaison des deux directions de recherche dans un seul algorithme a beaucoup d'avantages. C'est un algorithme incrémental, il accepte des exemples d'entraînement un par un, formant une généralisation locale partielle utilisable, même avec la possibilité qu'elle soit incomplète, après chaque exemple, contrairement aux algorithmes « *batch* » comme le ID3 ou tous les exemples d'entraînement doivent être présents pour le processus d'apprentissage. L'algorithme d'élimination de candidat pour l 'apprentissage du concept « *red ball* » est présenté dans la **figure 13**. Son illustration est dans la **figure 14.**

L'algorithme forme des groupes de concepts candidats: **G** le groupe de concepts généraux maximaux (*maximally general*), et **S** le groupe de concepts spécifiques maximaux (*maximally specific*). L'algorithme spécialise **G** et généralise **S** jusqu'à ce qu'ils convergent vers un concept commun. Cette combinaison de recherche dans deux directions dans un seul algorithme est efficace surtout à cause de son incrémentalité c.à.d qu'après chaque exemple, l'usager obtient une généralisation utilisable même si elle est incomplète.

Begin

*Initialize **G** to be the most general concept in the space;*

*Initialize **S** to the first positive training instance;*

For** each new positive instance **p

Begin

*Delete all members of **G** that fail to match **p**;*

***For** every **s** in **S, if s** does not match **p,** replace **s** with its most specific generalizations that match **p**;*

*Delete from **S** any hypothesis more general than some other hypothesis in **S**;*

*Delete from **S** any hypothesis not more specific than some hypothesis in **G**;*

End

For** each new negative instance **n

Begin

*Delete all members of **S** that match **n**;*

*For each **g** in **G** that matches **n,** replace **g** with its most general specializations that do not match **n**;*

*Delete from **G** any hypothesis more specific than some other hypothesis in **G**;*

*Delete from **G** any hypothesis more specific than some hypothesis in **S**;*

End

***If G=S** and both are singletons **then** the algorithm has found a single concept that is consistent with all data, the algorithm halts;*

*If **G** and **S** become empty **then** there is no concept that covers all positive instances and none of the negative instances;*

End

Figure 13: l'algorithme d'élimination de candidat

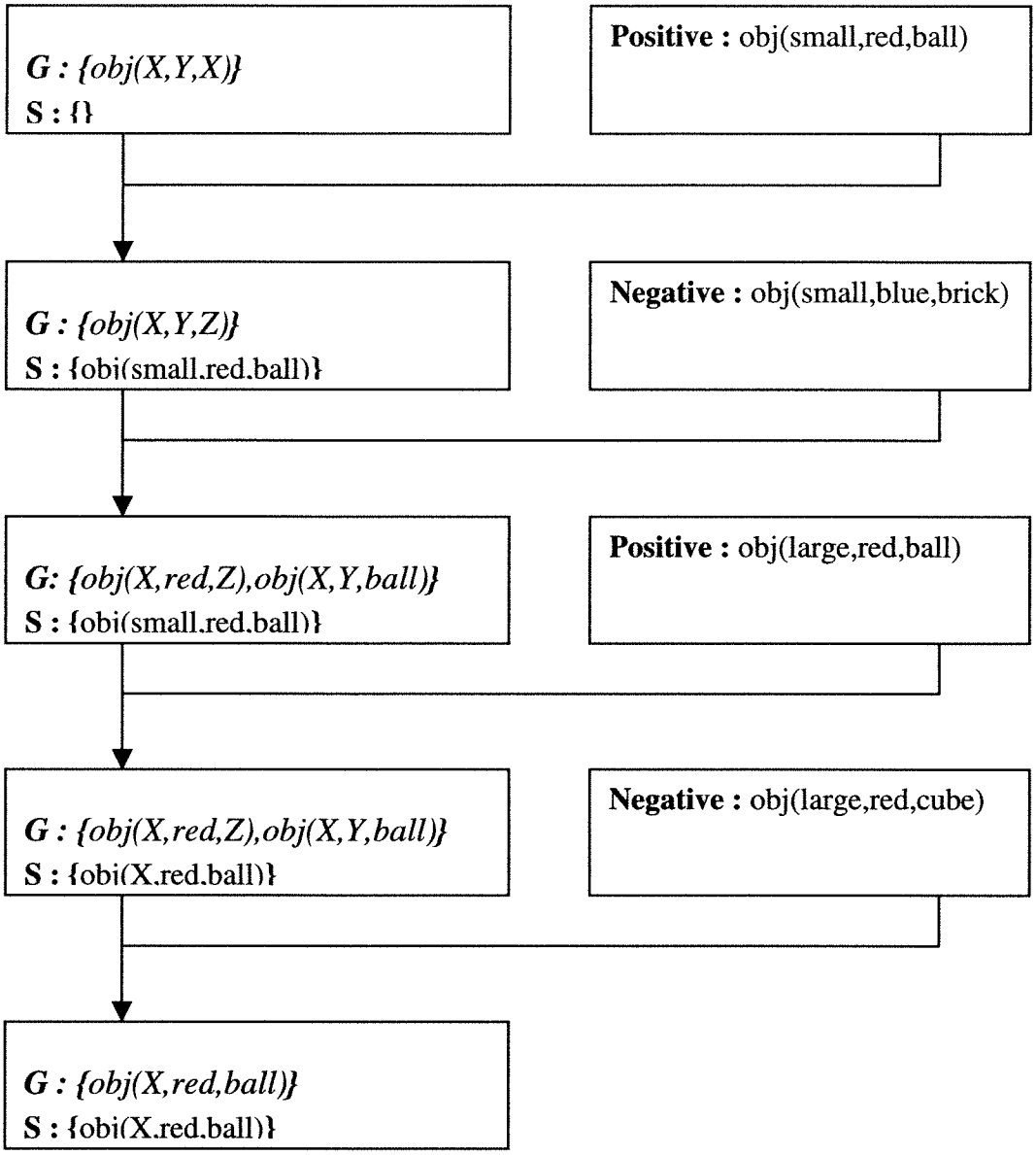


Figure 14: Le résultat de l'application de l'algorithme 11 sur l'espace de la figure 8

Dans cet algorithme, **G** est initialisé avec le concept le plus général alors que **S** porte le premier concept positif. Avec chaque concept négatif, les membres de **S** qui s'apparient avec le concept négatif sont enlevés, chaque élément de **G** qui s'apparie avec *n* est remplacé par sa spécialisation la plus générale, chaque hypothèse dans **G** plus spécifique d'une autre hypothèse dans **G** est supprimé ainsi que chaque hypothèse dans **G** plus spécifique qu'une hypothèse dans **S**. Dans le cas d'un concept positif, on supprime tous les membres de **G** qui ne s'apparient pas

avec le concept négatif. Pour chaque concept dans **S** qui s'apparie avec le concept positif on remplace le concept par la généralisation la plus spécifique qui s'apparie avec le concept positif. Chaque hypothèse dans **S** plus générale qu'une autre hypothèse dans **S** est supprimée ainsi que chaque hypothèse dans **S** qui n'est pas plus spécifique qu'une autre hypothèse dans **G**.

Un des problèmes majeurs de la recherche dans l'espace de version est l'augmentation rapide du volume de l'espace. L'autre est l'utilisation d'une recherche en largeur qui est souvent inefficace. Pour résoudre ces problèmes, des recherches heuristiques ou des *sauts* sont souvent utilisées. Une autre approche est l'introduction des « *biais inductifs* », pour réduire le volume de l'espace de concepts. De tels biais imposent des contraintes sur le langage utilisé pour la représentation des concepts (*Turney, 1999*). Les langages utilisant un biais, sont essentiels pour réduire la complexité de l'espace de concepts, mais ils peuvent être incapables de représenter le concept à apprendre. Notre algorithme utilisera à son tour un espace de version, les phrases des documents d'entraînement formeront cet espace. En plus le concept de la généralisation la plus spécifique sera adoptée dans l'algorithme de ***GlobSum*** pour former les patrons les plus importants après l'entraînement fait. Le Chapitre 4 détaillera ce processus.

Algorithme ID3, Arbre de Décision :

Comme l'algorithme d'élimination de candidat, l'algorithme ID3 induit les concepts à partir des exemples. Les caractéristiques les plus importantes dans ID3 sont la représentation compréhensible des connaissances apprises, la gestion de la complexité, les heuristiques utilisées pour la sélection des concepts candidats et la capacité à traiter les erreurs causées par les bruits (*Quinlan, 1986*). ID3 présente les concepts comme « *Arbre de Décision* », une représentation qui permet de déterminer la classification d'un objet en traitant ses valeurs pour certaines propriétés. On recherche à construire l'arbre de décision le plus simple pour classifier les exemples. Le but est de construire un arbre de décision qui soit

suffisamment général pour classifier les exemples et en même temps ignorer les contraintes non nécessaires. Plusieurs algorithmes avancés et optimisés du ID3 existent comme le CART (*Breiman, 1984*), on présente un simple algorithme ID3 (**figure 15**):

Function ***induce_tree****(example_set, Properties)*
begin
If *all entries in example_set are in the same class* ***then***
    return a leaf node labeled with that class;
Else
if *Properties is empty* ***then***
    return leaf node labeled with disjunction of all classes in example_set ;
Else
    Select a property, ***P****, and make it the root of the current tree;*
    Delete ***P*** *from Properties;*
    For *each value* ***V*** *of* ***P***
        Begin
        Create a branch of the tree labeled with ***V;***
        Let $\mathbf{partition}_v$ *be elements of example_set with values* ***V*** *for property* ***P;***
        Call ***induce_tree****(*$partition_v$*, Properties);*
        attach result to branch ***V;***
        End
End

Figure 15: l'algorithme ID3

L'exemple de classification du risque de crédit illustrera la construction de l'arbre de décision (**table 8**):

No.	Risk	Credit History	Debt	Collateral	Income
1.	High	Bad	High	None	$0 to $15k
2.	High	Unknown	High	None	$15 to $35k
3.	Moderate	Unknown	Low	None	$15 to $35k
4.	High	Unknown	Low	None	$0 to $15k
5.	Low	Unknown	Low	None	Over $35k
6.	Low	Unknown	Low	Adequate	Over $35k
7.	High	Bad	Low	None	$0 to $15k
8.	Moderate	Bad	Low	Adequate	Over $35k
9.	Low	Good	Low	None	Over $35k
10.	Low	Good	High	Adequate	Over $35k
11.	High	Good	High	None	$0 to $15k
12.	Moderate	Good	High	None	$15 to $35k
13.	Low	Good	High	None	Over $35k
14.	High	Bad	High	None	$15 to $35k

Table 8: Exemple de l'histoire de crédit

Un arbre de décision qui classifie les exemples de la table est présenté dans la **figure 15**. Il faut noter que cet arbre n'est pas unique, plusieurs « *autres arbres* » avec différentes profondeurs et complexité existent (*Quinlan, J.R, 1987*). Le choix de l'arbre optimal est élaboré dans une seconde étape. L'arbre de la **figure 16** n'utilise pas toutes les propriétés de la **table 8**, par exemple si une personne possède un bon crédit et peu de dette, on peut, par rapport à l'arbre, ignorer les autres attributs et le classifier comme « *peu de risque* ».

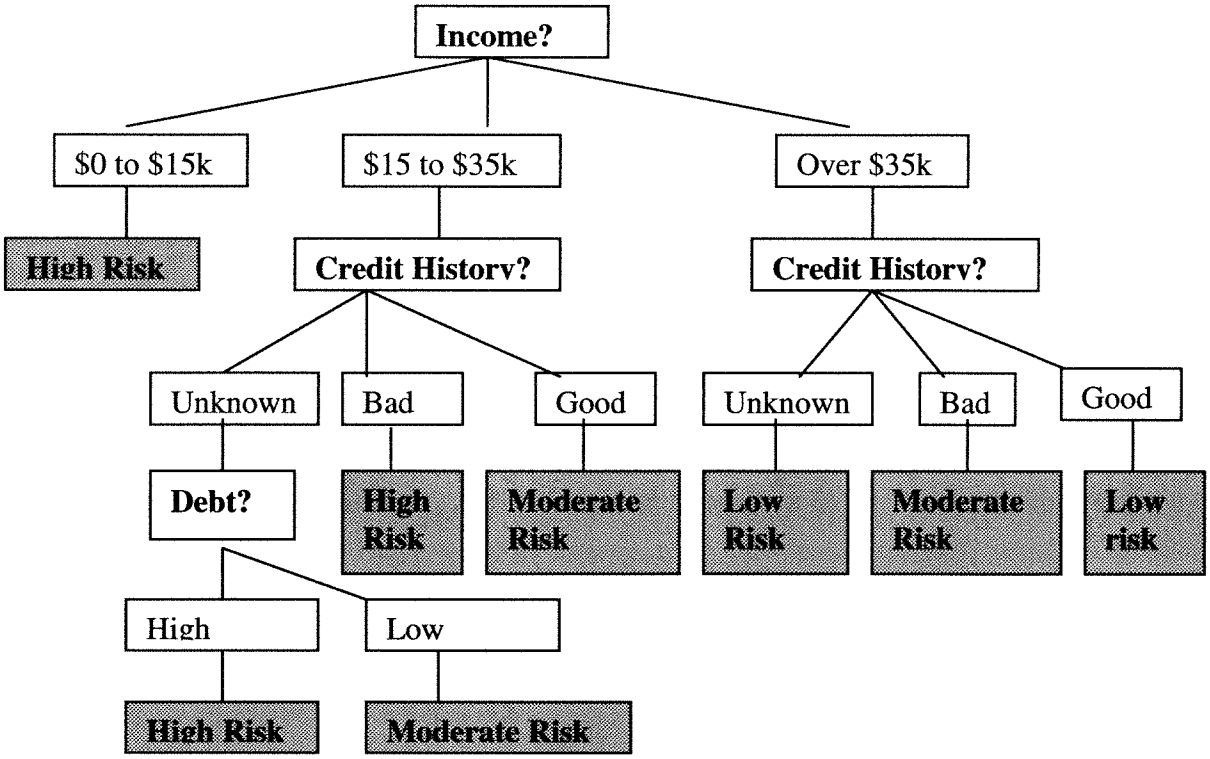


Figure 16: Arbre de décision pour la classification de crédit

Plusieurs arbres de décision sont possibles, le choix de l'arbre pertinent est effectué en utilisant le théorème de « *test d'information* » ou «*gain d'information* ». Chaque niveau de propriété apporte un certain gain d'information (*Shannon, 1948*), l'idée est de choisir la propriété ayant le plus grand gain. Mathématiquement le gain d'information d'une propriété P sur un ensemble d'exemple d'entraînement S formé des classes {C1,C2,...Cn} (Pour la table de la figure 8, les classes high risk, moderate risk et low risk) est défini par :

$$gain(S, P)=Entropie(S) - \sum_{1}^{n} [\ |Ci| / |C| * Entropie(Ci)\ ]$$ où :

i *représente les valeurs de l'attribut P (dans notre exemple, quand P est « credit history » alors i est {unknown, bad, good}*

|Ci| présente le cardinal de Ci (nombre d'élément dans Ci)

Gain(S,P) *est le gain de la propriété* ***P*** *recherchée*

Entropie(S) *est une mesure du degré de variabilité de tous les exemples avec* $Entropie(S)=\sum_{1}^{n} [\ -p(C_i)log_2(p(C_i))]$ *ou p(Ci) est la probabilité de Ci (fréquence relative de Ci dans S)*

Finalement on peut noter qu'il est possible de transformer chaque arbre de décision en un groupe de règles par conversion de chaque chemin possible dans l'arbre en une règle où la partie gauche est constituée de toutes les décisions prises jusqu'au nœud, et la partie droite est le nœud terminal.

Notre algorithme s'aidera des formations et arrangements des exemples dans les arbres de décision, la formation et les alignements des attributs, caractéristiques et des propriétés dans une table d'information. Il utilisera une table d'information avec une liste d'attributs similaire à la **table 8**.

Le « biais » inductif

Le biais inductif peut être défini par un ou plusieurs critères utilisés pour restreindre l'espace de concepts ou sélectionner certains concepts de l'espace. Les espaces d'apprentissage sont souvent grands; sans restrictions, la recherche de cet espace est difficile ou même impossible. Un critère est souvent nécessaire pour éliminer les généralisations excessives. Considérons par exemple l'espace des chaînes de caractères, avec les exemples positifs comme*{1100}* et *{1010}*, plusieurs généralisations existent, les chaînes commencent par un *1*, les chaînes terminant par un *0*, la conjonction des deux, la disjonction des deux...une restriction est nécessaire. L'algorithme ID3 utilise un biais dans son choix de l'arbre de décision, ce biais étant le gain d'information. L'autre type de biais est introduit par des contraintes syntaxiques sur la représentation des concepts appris.

Malgré les nouvelles recherches pour le développement d'un système qui change son biais selon les connaissances et les exemples (*Utgoff, 1986*), la majorité des programmes d'apprentissage utilisent un biais fixe. Plusieurs biais ont été explorés dans l'apprentissage (*Mitchell, 1980*):

1. ***Biais de conjonction*** : Les connaissances sont limitées par des conjonctions. C'est une bonne fonction pour limiter les généralisations.

2. ***Limitation du nombre de disjonctions*** : L'utilisation d'un nombre défini de disjonctions pour pouvoir représenter les connaissances et en même temps éliminer les généralisations excessives.
3. ***Vecteurs attributs*** : Une représentation qui décrit les objets comme un groupe d'attributs qui diffèrent d'un objet à l'autre.
4. ***Arbres de décision*** : Une représentation de concept dans l'algorithme ID3
5. ***Clauses de Horn*** : Une représentation pour la production de règles à partir d'exemples.

Certains programmes utilisent leurs propres connaissances du domaine pour produire des restrictions et des biais spécifiques, ces types de biais sont les biais sémantiques.

Le but des biais inductifs est de restreindre un groupe de concepts de façon à pouvoir chercher efficacement ce groupe et trouver une bonne qualité de définition des concepts. Le choix du biais idéal est souvent difficile à déterminer pour qu'il soit suffisamment grand pour inclure la solution du problème, et aussi suffisamment petit pour assurer une généralisation de l'espace de recherche (*Baxter, 2000*). ***GlobSum*** s'aidera d'un biais qui limitera l'espace de recherche. Le biais est un degré de similarité entre les phrases interprétées où toute phrase ayant un degré inférieur à une limite donnée sera rejetée.

Apprentissage par explication

L'apprentissage explicatif essaie de construire une explication des exemples d'entraînement, car un exemple est la suite logique d'une théorie. Une généralisation de l'explication trouvée est ensuite effectuée. Plusieurs algorithmes de généralisation explicative ont été développés, **STRIPS** (*Fikes et al. 1972*), **Meta-DENTRAL .**

Les algorithmes explicatifs sont formés des composantes suivantes (DeJong, Mooney, 1986):

1. *Le concept recherché :* La tâche est de découvrir une définition effective du concept. Le concept peut être une classification, un théorème à démontrer, un plan ou bien une heuristique pour un problème.
2. *Exemple d'entraînement :* Une instance du concept recherché
3. *Une théorie de domaine :* Un groupe de règles utilisées pour expliquer comment l'exemple d'entraînement est une instance du concept recherché.
4. *Critère opérationnel :* Une méthode pour décrire la forme des concepts.

Bien que les algorithmes EBL (*explanation based learning*) soient utilisés dans plusieurs systèmes, plusieurs recherches restent actives pour le développement de ce domaine. ***GlobSum*** s'identifie avec les algorithmes EBL en suivant le format des quatre points mentionnés, le concept recherché (dans notre cas les patrons), l'exemple d'entraînement qui s'identifie avec les documents du nouveau domaine, la théorie de domaine qui sont les règles à découvrir et le critère opérationnel.

Algorithmes Non-Supervisés

Les algorithmes présentés à la section précédente implémentaient une forme d'apprentissage supervisé supposant l'existence d'un instructeur, d'une fonction où d'une méthode de classification. L'apprentissage non-supervisé élimine le besoin d'instructeur et oblige l'apprentissage à prendre et évaluer lui-même ses propres concepts. Deux types d'algorithme non supervisés sont présentés bien qu'il existe plusieurs autres à mentionner comme le « Self Organizing Map » (SOM) (*Kohonen, 1995*) très utilisé pour l'analyse de données et permettant de cartographier en deux dimensions et de distinguer des groupes dans des ensembles de données, les « Vector Quantization » (VQ) (*Grossberg, 1976*), une méthode généralement qualifiée d'estimateur de densité non supervisé et permettant de retrouver des groupes sur un ensemble de données, de façon relativement similaire à un « *k-means algorithm.* »

Algorithmes de découverte

Les premiers travaux dans les algorithmes de découverte ont débuté avec AM (*Lenat, 1977*) en 1977 pour la dérivation des concepts mathématiques. Plusieurs autres systèmes ont été développés suite au succès de AM comme les fameux systèmes de EURISKO, IL, BACON et SCAVENGER de *Stubblefield*. Malgré l'importance du domaine de découverte, le progrès a été lent et ces algorithmes ont été utilisés surtout dans les domaines de chimie. Bien que ***GlobSum*** soit supervisé par les exemples d'entraînement, il effectue un repérage de patron automatique par comparaison d'attributs de phrases interprétées d'où sa nature de découverte.

Algorithmes de classification

Les algorithmes de classification organisent des objets en une hiérarchie de classes avec certaines qualités, comme la maximisation des similarités des objets d'une même classe. La taxonomie numérique est la méthode la plus connue dans les algorithmes de classification (*Holland, 1986*). Les objets sont représentés comme vecteurs d'attribut associés à des points dans un espace n-dimensionel (n étant le nombre d'attributs), la similarité entre deux objets est la distance euclidienne entre les deux points correspondants dans l'espace. L'algorithme de formation de catégories, utilisant la métrique euclidienne, est le suivant (**figure 17**) :

Repeat

Select the pair with the highest degree of similarity, make that pair a cluster

Define the features of the cluster as some function (ex: average), of the features of the component members and then replace the component objects with this cluster definition

Until *one object left*

Figure 17: Formation de catégories

Cet algorithme peut être étendu pour traiter les concepts symboliques et numériques. Le changement principal est l'évaluation de la similarité symbolique qui est maintenant accomplie par un calcul de proportion des attributs en commun. Par exemple avec trois objets :

Object1 = {small, red, rubber, ball},

Object2 = {small, blue, rubber, ball},

Object3 = {large, black, wooden, ball},

Les similarités sont:

Similarity(Object1, Object2)= ¾,

Similarity(Object1, Object3)= Similarity(Object2, Object3)= ¼.

Le désavantage de cet algorithme est son manque de considération sémantique car on prétend que tous les attributs possèdent la même importance. Plusieurs variations existent qui tiennent compte d'un calcul d'entropie des attributs selon leur importance. Plus d'information et d'exemples de systèmes classificateurs se trouve dans *(Reitman, 1978).* ***GlobSum*** utilisera la notion de vecteurs d'attributs pour présenter les informations, ainsi que la distance euclidienne pour calculer les similarités des phrases.

3.3 GlobSum, un algorithme de découverte supervisé

Plusieurs types d'algorithmes d'apprentissage ont été présentés dans ce chapitre. Même si chaque algorithme est unique, plusieurs idées et caractéristiques ont été dégagées et utilisées dans notre algorithme de découverte de patrons.

L'idée de *GlobSum* est simple : On va introduire un groupe de documents comme un groupe d'entraînement, les phrases de ces documents seront interprétés lexicalement, syntaxiquement et sémantiquement pour dégager des attributs (Concept, relation, voix, position...). Les phrases interprétées sont ensuite

introduites à l'algorithme de découverte qui va les comparer et former des groupes de phrases satisfaisant un ensemble d'attributs et de patrons communs. A chaque groupe de phrases sera associé un poids selon le nombre de phrases qu'il contient et l'entropie de ses attributs. Les meilleurs groupes de phrases avec les meilleurs « poids » vont être gardés, et leurs patrons correspondant vont être sauvegardés dans des templates vides dans le dictionnaire conceptuel de ***SumUm*** à la place des anciens templates produis par appariement manuel du corpus. ***SumUm*** utilisera ainsi ces patrons dans sa sélection de contenu.

GlobSum utilisera un espace de version (Comme l'algorithme *Find-S* et *Version Space Search*) formé des phrases interprétées. Il utilisera encore un biais pour limiter cet espace et va interpréter les documents d'entraînement sous forme de vecteurs d'attributs (Similaire à l'algorithme *ID3*).
Les détails de ***GlobSum*** sont présentés dans le chapitre suivant.

Chapitre 4

4 Intégration de l'algorithme GlobSum dans SumUM

La route pour la production d'un système de résumé automatique est un chemin rempli de difficultés et de limitations : la limitation à un domaine, la pertinence des sujets jugés comme principaux, la génération du résumé et le degré de complexité sont tous des problèmes bien connus dans le domaine de production des résumés automatiques.

Les évaluations de ***SumUm*** (*Saggion, 2000c*) ont démontré sa capacité à sélectionner les bons sujets. L'utilisation de plusieurs méthodes complexes d'extraction de phrases lui donnent ce pouvoir de localisation de pertinence mais sont aussi la cause de sa complexité. D'autre part, la génération de phrases dans ***SumUm*** tire avantage des règles de transformation pour ajouter une variation et un niveau de personnalisation aux résumés générés. Ces avantages de ***SumUm*** nous motivent à travailler sur sa plus grande limitation de domaine qui peut être traitée par l'intégration d'un algorithme de découverte de patrons automatique facilitant son passage à un autre domaine. Pour souligner la limitation de domaine, nous présentons une évaluation de ***SumUm*** que nous avons faite (avec les patrons de documents techniques) sur des documents « *Non technique* ». Notre algorithme ***GlobSum*** est ensuite présenté avec ses caractéristiques, son implémentation et ses étapes de développement avec les idées des algorithmes d'apprentissage du *chapitre 3*. Finalement, deux types d'évaluations sont présentés pour juger l'efficacité de ***GlobSum*** dont l'objectif est d'automatiser la découverte de patrons par entraînement. Si ***GlobSum*** annexé à ***SumUm*** fonctionne sans variations majeures dans son efficacité, ***GlobSum*** est considéré comme un succès, sinon une nouvelle approche devra être conçue. Deux évaluations différentes sont exécutées dans le *chapitre 5* pour tester la performance de ***GlobSum***.

4.1 Comportement de *SumUm* sur les documents non techniques

Des évaluations effectuées sur ***SumUm*** ont démontré la qualité de la sélection indicative dans le processus de production de résumés automatiques. ***SumUm*** a été comparé à plusieurs systèmes commerciaux et de recherches, aux résumés déjà produits par des experts languistiques humains. Les systèmes comparés étaient « *Microsoft'97 Summarizer* », « *n-STEIN* » et « *Extractor* ».

SumUm a été capable, dans toutes les évaluations, de produire de résumés techniques de qualité similaire et même meilleure (On peut voir une liste détaillée de ces tests dans *Saggion, 2000d*). Mais quel sera le comportement de ***SumUm*** sur des documents non techniques? Pour répondre à cette question, on a évalué ***SumUm*** sur huit documents non techniques présentés à l'**annexe D**. Les mesures d'évaluation standards de *rappel* et *précision* ont été utilisées. En assumant que:

- A est le groupe de texte que le système (***GlobSum*** et ***SumUm***) a extrait dans la production du résumé automatique;
- B est le groupe de textes pertinents du résumé automatique (le résumé actuel produit par l'expert humain);
- AB est le groupe de phrases pertinentes produites par le système (***GlobSum*** et ***SumUm***) trouvé par comparaison au résumé actuel;

Alors la mesure de la précision et le rappel sont obtenus par la formule suivante :

$$Précision = AB / A$$

$$rappel = AB / B$$

On a déjà noté que l'autre limitation de ***SumUm*** se trouve dans sa dépendance sur la structure du document, c.a.d chaque document doit contenir un titre, conclusion et des sections. Pour simuler la structure d'un document technique et satisfaire cette limitation, les huit documents non-techniques (Ne possédant aucune structuration)

ont été segmentés à la main pour procéder à leurs évaluations (Cette limitation de structure peut être résolue en adoptant une structuration par paragraphe, l'idée importante ici est que cette limitation n'est pas difficile à résoudre comme la découverte de patrons).

Notre première évaluation calcule la précision et le rappel sur ces documents non techniques. Les documents, leurs résumés professionnels et les résumés respectifs produits par ***SumUM*** sont utilisés dans le calcul de la précision et du rappel. Les résultats sont présentés dans la **table 9**. La précision de ***SumUm*** sur les documents non techniques s'est dégradée de façon importante. Le rappel reste plus ou moins stable.

Numéro de l'article	**SumUm**	
	Précision	Rappel
1	.01	.20
2	.04	.19
3	0	.15
4	.02	.23
5	.03	.27
6	.03	.24
7	.01	.22
8	.07	.18
Average	.03	0.22

Table 9: Précision et rappel sur les documents non techniques

La seconde évaluation a porté sur le facteur humain. Six personnes ont consulté les huit articles, leurs résumés, et les résumés produits par ***SumUm***. Elles ont ensuite attribué une valeur entre 1 et 5, selon leur appréciation de la qualité du résumé produit par ***SumUm***. Chaque document a été aussi jugé comme indicatif ou non par ces personnes. Les résultats sont présentés à la **table 10**.

Numéro de l'article	SumUm	
	Indicatif?	Qualité
1	Non	1.2
2	Non	1.8
3	Non	2.0
4	oui	2.6
5	Non	1.2
6	Non	1.3
7	Non	1.1
8	Non	1.0
Moyenne	20%	1.85

Table 10: Jugements de la qualité et degré indicatif pour les documents non techniques

Les résultats des **tables 9** et **10** illustrent la faiblesse de ***SumUm*** sur des documents non techniques. 20% seulement des documents sont jugés comme indicatifs avec une qualité faible (1.85 par moyen). La cause de cette baisse de performance et de pertinence lors d'un changement de domaine peut être expliquée par le choix des patrons et du style utilisé par l'auteur dans chaque domaine. Plus spécifiquement, lors de la sélection indicative, ***SumUm*** applique les patrons du dictionnaire conceptuel sur les phrases interprétées pour remplir les templates convenables. Cependant, dans le cas des documents non techniques, très peu de template sont remplis puisque tout simplement on ne trouve pas de phrases où, par exemple, le patron de cooccurrence du concept « *author* » et la relation « *make known* » (skip1+Author+Make Known+XX+eos) s'applique. Dans les plus mauvais cas, des templates totalement non pertinents peuvent être remplis à cause d'erreur de discours (Les problèmes de discours se trouvent toujours dans ***SumUm*** puisque les concept et relation sont identifiés par des termes lexicaux qui peuvent avoir différentes interprétations selon le discours). En conclusion, la dégradation de performance de ***SumUm*** est presque totalement causée par les mauvais patrons puisque quelque soit la méthode de sélection de contenu que ***SumUm*** va utiliser à

la fin, elle va être sur les templates. Il est donc clair, que l'automatisation du processus de localisation de patrons est une étape majeure dans la facilitation du passage entre domaines dans ***SumUm***. Cette automatisation va être testée sur les documents techniques et non techniques. Nous présentons maintenant notre algorithme, ses caractéristiques et ses attributs. Nous décrirons ensuite les fonctions que nous avons implémentées et les tables que nous avons construites.

4.2 Structure de GlobSum

Pour produire un résumé, ***SumUm*** débute par la segmentation du document introduit par sections par un programme externe. Le tagger est appliqué pour associer des catégories lexicales à chaque phrase, des automates dégagent la structure syntaxique et le dictionnaire conceptuel est consulté dans une première passe pour voir les concepts et les relations et les associer aux phrases pour une interprétation sémantique. Jusqu'ici les patrons des templates du dictionnaire conceptuel ne sont pas encore utilisés. Après l'interprétation, ***SumUm*** consulte le dictionnaire conceptuel dans une deuxième passe, applique les patrons et remplit les templates convenablement qui sont ensuite choisis dans une dernière étape pour produire le résumé. ***GlobSum*** exécutera ce même scénario. La seule différence etant que les patrons des templates du dictionnaire conceptuel seront obtenus par un algorithme d'entraînement et de découverte et non pas par appariement manuel de corpus. Ces patrons, et par suite leurs templates, peuvent changer en fonction des documents utilisés dans l'entraînement. Le nouveau scénario est le suivant :

- Entraîner avec ***GlobSum*** et découvrir les nouveaux patrons
- Structurer les patrons dans les templates et sauvegarder ces nouveaux templates dans le dictionnaire conceptuel
- Exécuter ***SumUm*** avec le nouveau dictionnaire conceptuel

Dans sa publication « The Essential Guide for Writers, Editors and Publishers » John Grossman (Grossman, 1993) indique la différence entre les extraits de textes de domaines différents indiquant que chaque auteur utilise « *sans le savoir* » son propre style qui, malgré qu'il soit personnel, reste similaire et conforme aux styles des autres auteurs du même domaine. Pour chaque style de texte, il existe des règles non documentées (unwritten rules) que chaque auteur suit sans le réaliser. En écrivant des articles de documentations de logiciel, les auteurs (dans ce cas des informaticiens) utiliseront des termes, des phrases qui se répéteront dans les documentations de logiciel (*Afnor, 1984*). Les textes dans le journal de *CNN* sur le tremblement de terre en Inde, la famine en Afghanistan ou l'éruption de volcan au Japon contiendront des termes comme « *crise* », « *mort* » « *destruction* » et « *aide* », alors que les textes qui détaillent les arts contemporains utiliseront un style différent, des concepts et relations différents, des termes comme « *arts* », « *artiste* » mais sûrement pas des termes comme « *crise* » et « *mort* » qui sont abondent dans le domaine des crises internationales. C'est un point important puisque les concepts de « *mort* » et de « *crise* » existent, mais ils n'ont pas la même « *importance* » ou occurrence dans le domaine des arts.

GlobSum se base sur l'idée que les concepts et les relations sont constants et ne changent pas selon le type de texte, alors que le degré d'apparition, la fréquence et la cooccurrence de ces concepts et relations est la différence clé entre les textes d'un domaine et l'autre comme on l'a déjà démontré. Pendant le développement de ***SumUM,*** le style des documents techniques a été dégagé par consultation de corpus et le dégagement des cooccurrences de concepts et de relations particulières appelées patrons. Nous allons automatiser ce processus pour découvrir automatiquement les patrons, et ainsi pouvoir réutiliser ce processus dans un autre domaine pour dégager le style de ce nouveau domaine, une étape importante pour enlever la limitation de domaine.

Le développement de ***GlobSum*** a suivi les étapes suivantes :

- Entraînement : Cette phase est divisée en trois tâches principales :

- Interprétation de phrases. Lexical par un tagger, syntaxique par les automates (FST) et sémantique par consultation des concepts et relations. Le dictionnaire conceptuel contient la liste de concepts et relations. Pas de templates (patron) existent à ce moment.

- Dégagement des attributs de chaque phrase: les concepts, les relations, la voix et la position.

- Construction de la table de connaissance des attributs (*TCA*) constituée des phrases interprétées et leurs attributs. Le TCA formera notre espace de version.

- Étude de la base de connaissance des attributs, application d'un biais indicatif pour limiter notre espace de version.
- Groupement et « clustering » des phrases du TCA selon leurs attributs et les règles les plus spécifiques (*Maximally specific*). Formation de la table de groupes et de patrons préliminaire constituée des groupes découverts.
- Filtrage de la table de groupes selon plusieurs critères pour former les patrons finaux.
- Structuration des patrons découverts dans des templates vides et sauvegarde dans le dictionnaire conceptuel.
- Entraînement fini. Nouveau dictionnaire conceptuel disponible. Utiliser ce dictionnaire avec ***SumUm*** pour produire de résumés techniques.

La **figure 18** présente la structure de ***GlobSum***. La phase d'entraînement débute par une étape d'introduction d'une collection de 40 documents du corpus à l'interprétateur de phrases (La liste complète de ces documents se trouve à **Annexe F**). Dans cette phase les phrases sont interprétées lexicalement, syntaxiquement et sémantiquement : Un tagger lexical est d'abord utilisé pour dégager les catégories lexicales. On a utilisé un programme externe développé à l'Université de Montréal, le même qui a été utilisé dans ***SumUm*** (*Saggion, 2000d*). Un exemple d'une phrase et l'interprétation lexicale correspondante est présenté à la **figure 19**. Les catégories lexicales du tagger sont combinées avec le genre, temps, nombre, type etc.. pour un total de 214 catégories.

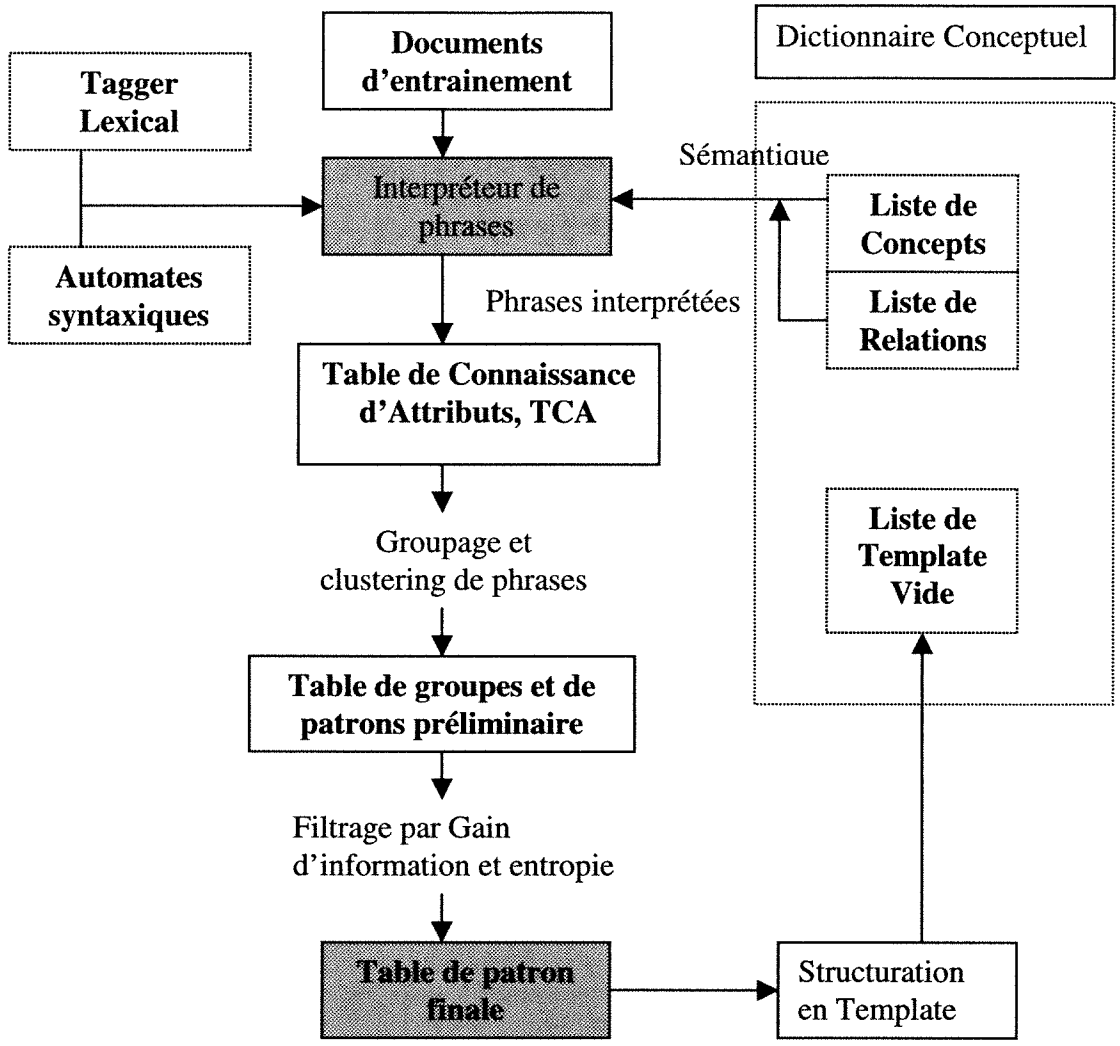


Figure 18: La structure de l'algorithme de découverte de patrons

Phrase d'entrée	**Tagger Lexical**
Introduction	Introduction : NomC-sing/introduction
	{para}
The automatic	The : Dete-dart-sgpl-def/a
Control	Automatic : AdjQ/automatic
Departement	Control : Nomc-sing/control
…	Departement : Nomc-sing/departement
…	…
	{EOF}

Figure 19: Le tagger Lexical

la structure syntaxique est déterminée en utilisant des grammaires compilées et des automates. On ne va pas détailler cette section puisqu'on utilise la même grammaire utilisée dans ***SumUm*** (*Saggion, 2000d*); il suffit de mentionner que cette grammaire est représentée par les FST (Finite state transducers) et que les phrases sont récursivement passées aux FST pour déterminer leurs syntaxes et les présentés en des structures *Prolog*. La phrase de la **figure 19** est présentée à la **figure 20** après l'interprétation syntaxique (GN4 est juste une référence au FST qui représente la partie de la grammaire : Dete A+ N+).

The automatic control departement (Dete A+ N+, Determinant Adjective group Noun Group)
(syncat, gn), (gntype, GN4), (string, [The, automatic, control, departement]), (canon, [automatic, control, departement]), (DeteType, dart), (type, def), (Nbr, sing)

Figure 20: L'interprétation syntaxique

Enfin le dictionnaire conceptuel ou plus spécifiquement la liste de concepts et relations est consultée dans une première passe pour associer les sémantiques convenables aux phrases. On a utilisé les listes de concepts et de relations qui se trouvent à l'annexe **A et B**.

Dans notre exemple, la phrase « *the automatic control departement* » le concept « *institution* » est identifié à cause du nom commun « departement ». Désormais aucune étude de discours n'est effectuée pour assurer l'exactitude du concept et relation, juste un simple test de syntaxe est effectué pour diminuer le bruit : un concept doit être un nom commun ou un pronom, une relation doit être un verbe. **Figure 21** présente la phrase interprétée sémantiquement.

The automatic control departement (Dete A+ N+, Determinant Adjective group Noun Group)
(syncat, gn), (gntype, GN4), ***(Concept, Institution)***, (string, [The, automatic, control, departement]), (canon, [automatic, control, departement]), (DeteType, dart), (type, def), (Nbr, sing)

Figure 21: Interprétation sémantique

Dans la seconde étape, plusieurs attributs des phrases interprétées des documents d'entraînement sont structurés avec les phrases correspondantes dans une table appelée table de connaissance d'attributs ou TCA. Quatre types d'attributs sont dégagés des phrases interprétées:

- ***La liste de concepts :*** On va extraire pour chaque phrase une liste de concepts qu'elle contient. C'est une tache simple puisque les concepts ont été déjà associés aux phrases lors de l'interprétation sémantique. On associe à chaque concept un poids qui est sa probabilité sur toutes les phrases du corpus d'entraînement. Si on dispose de 2000 phrases interprétées où le concept « author » apparaît 200 fois, alors on associera le poids de 200/2000=0.1 pour le concept « author ». Cette mesure permettra de nous offrir une idée de l'importance de chaque concept. Ce poids est cependant successible au bruit de discours puisque chaque concept est après tout définie par des termes lexicaux..
- ***La liste de relations :*** La liste de relations est également dégagée pour chaque phrase interprétée. C'est le même processus que le dégagement de la liste de concepts. Chaque relation est aussi attribuée un poids.
- ***Voix de la phrase :*** Un procédure est implémenté pour examiner les verbes et déterminer leurs voix : passive ou active. La voix est importante puisqu'elle détermine en partie la représentation du patron plus tard. Comme on a vu au **chapitre 2**, la phrase « on a présenté la recherche en profondeur… » s'unifie avec le patron « SKIP1+Author+Make Known+XX+eos » seulement quand la voix est active. Dans le cas où la voix est passive, XX (qui se remplira convenablement lors de la sélection) se trouve avant la relation. Dans notre cas

la phrase « La recherche en profondeur a été présentée... » aura le patron « XX+Make Known+SKIP1 ».

- ***Position de la phrase :*** L'interpréteur examine et sauvegarde la position de la phrase sous forme :

  « Nom du document et numéro de la session, Numéro de phrase dans la session ». Par exemple (zak.section.1,2) indique la deuxième phrase de la première session du document intitulé « zak » alors que (zak.abstract,1) indique la première phrase de l'abstract du document « zak ».

Ces attributs sont sauvegardés dans la *TCA* avec les phrases complètement interprétées (Résultat de notre interpréteur de phrases), la position de la phrase est utilisée comme clé de la table. **Table 11** illustre quelques entrées de la TCA.

Position Format : (Fichier, Phrase)	**Phrase interprétée en forme de liste Prolog**	**Liste de concepts (Concept, poids)**	**Liste de relation (Relation, poids)**	**Voix**
(zak.abstract,1)	[...]	(author,0.1), (research paper,0.07)	(make known, 0.8)	Active
(zak.abstract,2)	[...]	(Summary,0.03)	()	Passive
(zak.section.1,1)	[...]	()	(Focus,0.02), (define, 0.023)	Active
(zak.section.1,2)	[...]	...	...	.....
...	[...]	...	...	...
...	[...]	...	...	...
(sun.section.4.2,3)	[...]	()	()	Active
(sun.section.5,1)	[...]	(objective,0.1), (mathematical,0.06), (captionning, 0.06)	(Essential,0.02), (infer(0.032)	Active

Table 11: Table de connaissances d'attributs

En examinant la **table 11**, en peut voir que la première phrase de l'abstract du document « *zak* » (*zak.abstract,1*), en voix active, contient la cooccurrence des concepts « *author* », « *research paper* » et la relation « *make known* ».avec leurs poids respectifs. L'attribut « **Phrase interprétée en forme de liste Prolog** » contient la phrase complète interprétée comme celle de la **figure 11**. (cette phrase est très grande pour la mettre dans la **table 11**). La l'algorithme de construction de la TCA est présenté ci dessous :

Initialize TCA to [ ]

For each Document Xi in our training corpus (X1,X2,...Xn)

Begin

For each phrase Y in Xi

Begin

Use the Lexical tagger to interpret Y, output phrase Y1 lexically interpreted;

Use the grammars and automata (FST) on Y1 for syntaxical interpretation, output Y2 syntaxically interpreted;

Use the conceptual dictionary to associate concepts and relations to Y2, output Y3 semantically interpreted;

Extract all concepts in Y3, form the concept list;

Extract all relations in Y3, form the relation list;

Extract the voice and position of Y3;

Add Y3, Concept list, Relation list, voice and position to TCA;

End

End

Output TCA;

La TCA contient ainsi notre espace de version prêt à être manipuler par l'algorithme de découverte qui formera des groupes de patron. Mais une question se pose : Pourquoi utilise t-on dans la TCA toutes les phrases des documents d'entraînement et non pas seulement les phrases qui ont été utilisées dans les résumés professionnels? La raison est la suivante :

Les résumés professionnels ne sont pas toujours disponibles pour tous les documents et même s'ils étaient, ils peuvent être souvent totalement transformés par les résumeurs que les patrons sont totalement perdus. Dans le contexte d'un document qui explique la recherche en largeur, la phrase « The author was able to present the breadth first algorithm... » du document peut être présentée par un

résumeur comme « Breadth-first, is it worth it? » dans le résumé professionnel perdant ainsi notre patron. Ces deux raisons nous ont poussé à travailler avec toutes les phrases du corpus d'entraînement dans la TCA.

Il est probable cependant, que certaines phrases interprétées ne contiennent ni de concepts ni de relations correspondant à des attributs vides dans le TCA, ce point constituera notre biais indicatif : Chaque phrase contenant une liste de concepts vides **et** une liste de relations vides sera éliminée du TCA comme l'entrée (sun.section.4.2,3) de la TCA de la **table 11**. Ces phrases contiennent ou bien des termes lexicaux hors de la portée du taggeur (Le tagger à une exactitude de 95%) ou bien des termes qui ne sont identifiés avec aucun concept ou relation (Les concepts et relations sont identifiés par des termes lexicaux déterminés).

Le but de la seconde phase est d'arriver à partir de la table de connaissance d'attributs à des groupes de phrases qui satisfont le même patron. Pour chaque phrase $x(a_1\ldots,a_n)$ (La phrase x avec les attributs a_i) de la TCA, on va former un groupe $GP(a_1\ldots,a_n)$ contenant initialement la phrase $x(a_1\ldots,a_n)$. On va comparer ensuite $x(a_1\ldots,a_n)$ à chaque autre phrase $y(b_1,\ldots,b_n)$ dans la TCA. A ce moment trois choses peuvent arriver :

- x=y, c.a.d a1=b1,...an=bn, dans ce cas on ajoute y à $GP(a_1\ldots,a_n)$.
- $x(a_1\ldots,a_n) > y(b_1,\ldots,b_n)$. Dans ce cas les attributs de y sont plus spécifiques que ceux de x. $x(a_1\ldots,a_n) > y(b_1,\ldots,b_n)$ si et seulement si les contraintes suivantes sont satisfaite : Premièrement l'attribut correspondant à la liste de concepts de x est plus général que/incluse dans la liste de concept de y ([author] $\subseteq$ [author, institution]). Et deuxièmement l'attribut correspondant à la liste de relation de x est plus général que/incluse dans la liste de relation de y ([make known] $\subseteq$ [make known, focus]). Pas de contraintes sont associées à la voix ou la position. Le raisonnement est le suivant : Si la phrase « the author present ... » apparaît comme la première

phrase d'une section d'un document et comme la troisième phrase d'une section d'un autre document on va la considérer comme un patron indépendamment de la position. On pourrait à la limite mettre la contrainte que la position de x doit être « en voisinage » (3 ou 4 phrases de moins ou de plus) de la position de y, mais actuellement nous n'avons pas considéré ces contraintes.

- Si aucune des conditions précédentes n'est satisfaite alors on passe à la valeur suivante de la TCA.

A première vue, on peut se demander pourquoi on compare toutes les phrases entre elles. La réponse est qu'après observation de corpus, on s'est aperçu que plusieurs phrases peuvent se répéter dans le même document. « the author present … » peut être mentionné dans le début d'un document ainsi que dans la dernière section du même document, et en comparant les phrases exclusivement de chaque document on risque de perdre des informations importantes. C'est pourquoi, si les deux premiers documents de l'entraînement étaient « zak » et « sun », on compare la phrase 1 du document « zak » directement avec la phrase 2 de ce document et n'on pas à la phrase 1 du prochain document de l'entraînement « sun ». L'algorithme de groupage est présenté à la **figure 22**. Le résultat final est la table de groupes contenant tous les groupes de patrons possibles.

```
Table of groups is empty
For each x(a1,...,an) in the TCA
    If (a1,...,an) is already represented in the table of groups as
    GP(a1,...,an) then stop and go to the next x(a1,...an)
    If not then
        Begin
        Create a group GP(a1,...an) and initialise it to
        x(a1,...,an)
            Begin
            For each y(b1,...,bn) in the TCA
                If x(a1,...,an) = y(b1,...,bn) or
                x(a1,...,an)>y(b1,..,bn) then add
                y(b1,...,bn) to GP(a1,...,an)
            End
        Add GP(a1,...,an) to table of groups
        End
End
```

Figure 22: L'algorithme de groupage

L'algorithme de la **figure 22** débute par une table de groupe et de patrons vide. L'algorithme examine la TCA et pour chaque phrases de la TCA possédant les attributs (a1,…,an) identifiant un patron, l'algorithme va former un groupe de ce patron intitulé GP(a1,..,an) contenant tous les autres phrases de la TCA qui satisfont ce patron (les phrases ayant le même patron/attributs ou un plus spécifique). Ce groupe est ensuite ajouté à la table de groupes et de patrons et le processus est recommencé pour la phrase suivante de la TCA après avoir vérifié si un groupe vérifiant le patron de cette phrase n'existe pas déjà.

Voici un exemple :

Si la première phrase de la TCA était « The autheur presents a look on the breadth-first algorithm ». la liste d'attributs de cette phrase sera

a1=section1, phrase 3 (par exemple), a2=[author], a3=[make known], a4=active. Alors l'algorithme de groupage va trouver toutes les phrases de tous les documents d'entraînement qui ont ce patron exact ou un patron plus spécifique comme la phrase « we describe in this paper the impact of nuclear fusion » où a2=[author, research paper] et a3=[make known] (la phrase 1 est plus grande la phrase 2 selon notre définition). Ces phrases seront groupées dans GP([author],[make known]). Quand l'algorithme rencontre une autre phrase ayant exactement les attributs a2=[author] et a3=[make known] il va sauter à la phrase suivante du TCA (puisque à ce moment GP([author],[make known]) existe déjà dans la table de groupes). On note ici, qu'au fur et à mesure, l'algorithme va former le groupe GP([author, research paper],[make known]) correspondant à la deuxième phrase de l'exemple. La table de groupes et de patrons contiendra ainsi tous les groupes possibles dans le corpus de l'entraînement. La tâche maintenant est de filtrer ceux qui nous intéressent pour les considérer comme nos patrons finaux. Après tout GP([author],[make known]) peut être facilement transformé en « skip1+author+make known+XX » et s'intégrer à un template.

Table 12 illustre un exemple d'une table de groupes et de patrons. On a ajouté une colonne indiquant le nombre de phrases qui satisfont un groupe. La voix et la position ne sont pas inclus dans cette table mais ils peuvent être dégagées directement des phrases de la « liste de phrases ».

Groupe	Liste des phrases	Nombre de Phrases
GP([Make Known, Investigate], [Research Paper, Author])	[(zak.section.1,1), (info.section.2,2),...]	95
GP([Make Known, present], [focus, institution])	[(zak.title,1), (nm.abstract,2)...]	8
GP([Author], [Interest])	[(nm.section.4,1), (lorc.section.3.1,1)..]	86
...	...	...

Table 12: Table de groupes et de patrons

L'étape suivante consiste d'un filtrage de la table de groupes et de patrons pour choisir les groupes les plus importants et éliminer les autres. Pour cela on va se baser sur l'étude de corpus qui a été faite par Saggion (*Saggion, Lapalme 1998*) lors du développement de ***SumUm***. On a déjà présenté une table résumant les observations de Saggion, la **table 3** du **chapitre 2**. Cette table cite que 35% des phrases pertinentes proviennent des titres et conclusions des sections et 30% contiennent des expressions indicatives. Dans notre table de groupes et de patrons, on va utiliser d'abord le nombre de phrases de chaque groupe comme un premier filtre variable. Après tout, on peut facilement supposer que les groupes contenant un petit nombre de phrases, c.a.d satisfaits par peu de phrases sont à rejeter. On va avoir une valeur de « seuil » tel que chaque groupe ayant un nombre de phrases plus grand que ce seuil sera considéré comme patron et les autres seront à rejeter. On va tester le comportement de ***GlobSum*** avec différentes valeurs du seuil : Un très grand seuil limitera les patrons qu'on va avoir (on ne va pas avoir beaucoup de patrons) résultant en des résumés qui manquent souvent beaucoup d'idées principales, alors qu'un très grand seuil (Beaucoup de patrons seront gardés) résultera en des résumés trop vagues. Les tests sur des différentes valeurs du seuil sont présentés dans le **chapitre 5**.

Pour le second filtre, on examinera chaque groupe pour voir les positions des phrases de ce groupe. Si la majorité des phrases proviennent des titres et conclusion des sessions on va diminuer le seuil de ce groupe, sinon le seuil restera le même. De cette façon on tient compte de l'importance des phrases provenant des titres et des conclusions des sessions. Le résultat de filtrage de la table de groupes est une table qui contient nos patrons.

Une fois la table de groupes et de patrons filtrée, on passe à la dernière étape de notre algorithme qui s'occupe de la transformation de ces patrons en templates qui seront à leur tour insérés dans le dictionnaire conceptuel. La structure d'un template est présentée ci dessous :

Id	3
Pattern	Skip1+Author+Make Known+XX+eos.
Positions	Section 1, Last Section
Topic Candidates	
Weight	

Dans cette dernière étape on va associer à chaque groupe de patrons filtrés un nombre unique qui va être un le « **Id** » du template. On va aussi dégager la liste de positions des phrases de ce groupe et l'insérer dans le slot « **Position** ». Le slot « weight » reste toujours vide et va être rempli pendant la phase de sélection comme on a mentionné au **chapitre 2.** Le slot « Topic Candidates » va être rempli par la valeur de XX du patron lors de la sélection. Ce qui reste alors est le « **pattern** ». Pour former le « **pattern** » on va prendre les attributs $a_1,\ldots,a_n$ du groupe $GP(a_1,\ldots,a_n)$, on va consulter la valeur de l'attributs « voix », si la voix est active, notre variable XX se trouvera avant la relation, sinon elle se trouvera après la relation. Quand la voix est active XX sera considéré comme le complément de la relation, quand la voix est passive XX sera le sujet. On va aussi insérer des « skip » dans tous autres positions. La fin de la phrase sera noté par un « eos ». Par

exemple, en prenant le groupe GP([author],[make known]) : L'attribut voix est consulté et en supposant que la voix est active, XX sera le complément direct du verbe identifiant la relation « make known » et notre patron sera, après l'insertion convenable des « skip », « Skip1 + Author + Skip2 + make known + skip3 + XX + skip4 + eos ». La phrase « the author presents a breadth first algorithm. » sera appliquée à notre patron avec Skip1=[], Skip2=[], skip3=[], XX=[a, breadth, first, algorithm], skip4=[] et le point à eos. Avec ce processus on obtiendra le « slot » « **pattern** » de notre template.

Les templates vides ainsi formés seront sauvegardés dans le dictionnaire conceptuel avec la liste de concepts et relations pour être utilisé par ***SumUm.*** La structure et le placement de ***GlobSum*** par rapport à ***SumUm*** est présenté à la **figure 23.**

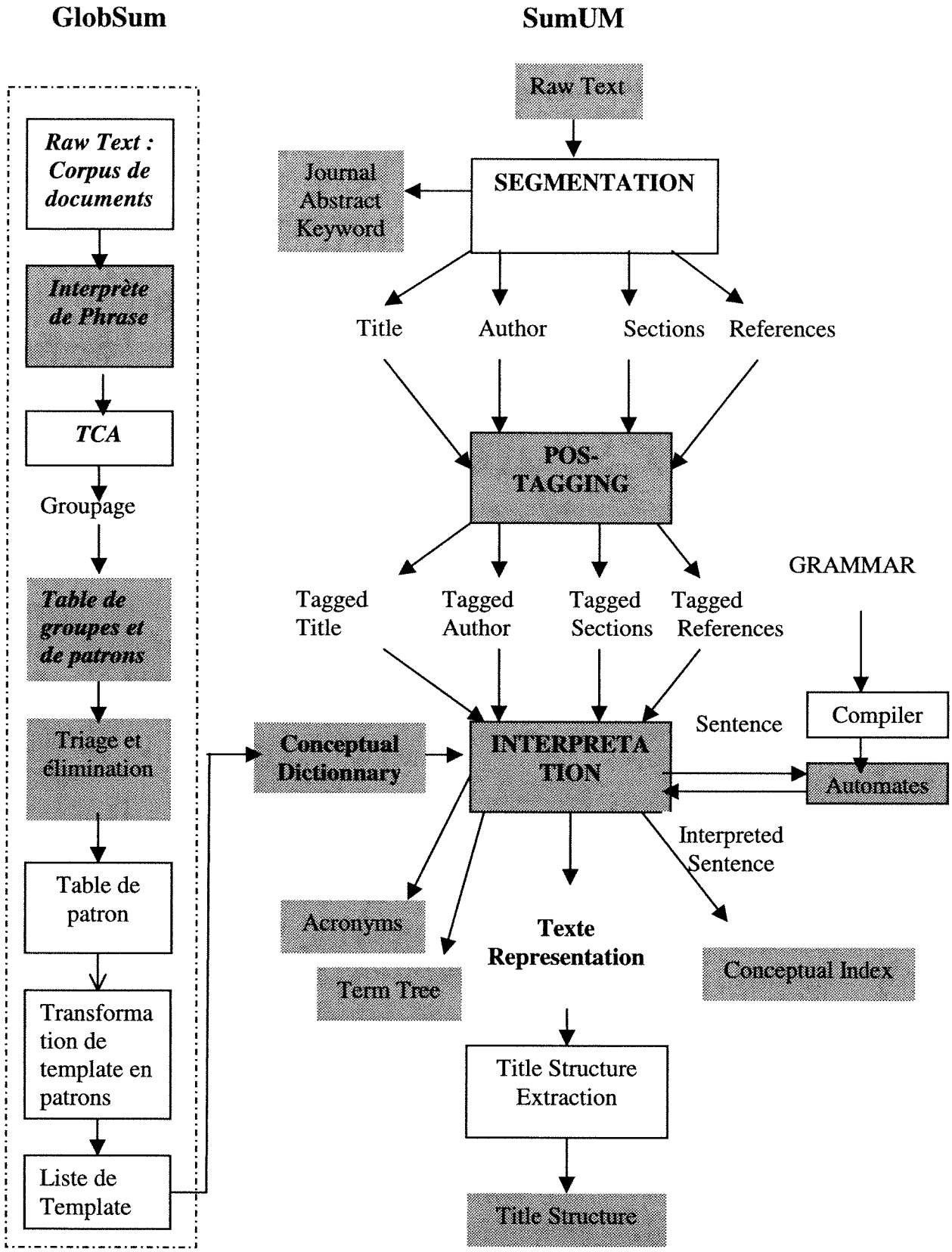


Figure 23: L'intégration de SumUm et GlobSum

La **figure 23** montre l'integration de ***GlobSum*** à ***SumUm***. La sortie finale de ***GlobSum*** est la liste de templates qui est intégrée dans le dictionnaire conceptuel de ***SumUm***. **Figure 23** montre l'interprétation seulement dans **SumUm** (pas de sélection dans cette figure).

L'intégration de l'interface de ***GlobSum*** est présentée à la **figure 24**. ***SumUm*** contient normalement les options 0 à 4 qui débute par une compilation d'automates qui transforme les grammaires (FST) en des listes Prolog (option 0), une analyse d'un document technique (interprétation, option 2) puis par la sélection (Summarizing, option 4). Les options de ***GlobSum*** sont les 5 au 9 : l'option 5 sert pour voir la liste de documents dans notre corpus d'entraînement, l'option 6 interprète les phrases des documents d'entraînement et forme la TCA. L'option 7 applique l'algorithme de groupage et de filtrage. L'option 8 liste les patrons trouvés pour le seuil utilisé et l'option 9 transforme les patrons en templates et les sauvegarde dans le dictionnaire conceptuel. Chaque document prend de 5 à 7 minutes dans son interprétation dépendamment de sa longueur, soit entre 3 heures et 4 heures pour les 40 documents de l'entraînement. La formation de groupes prend entre 10 et 20 minutes dépendamment de la grandeur de la TCA.

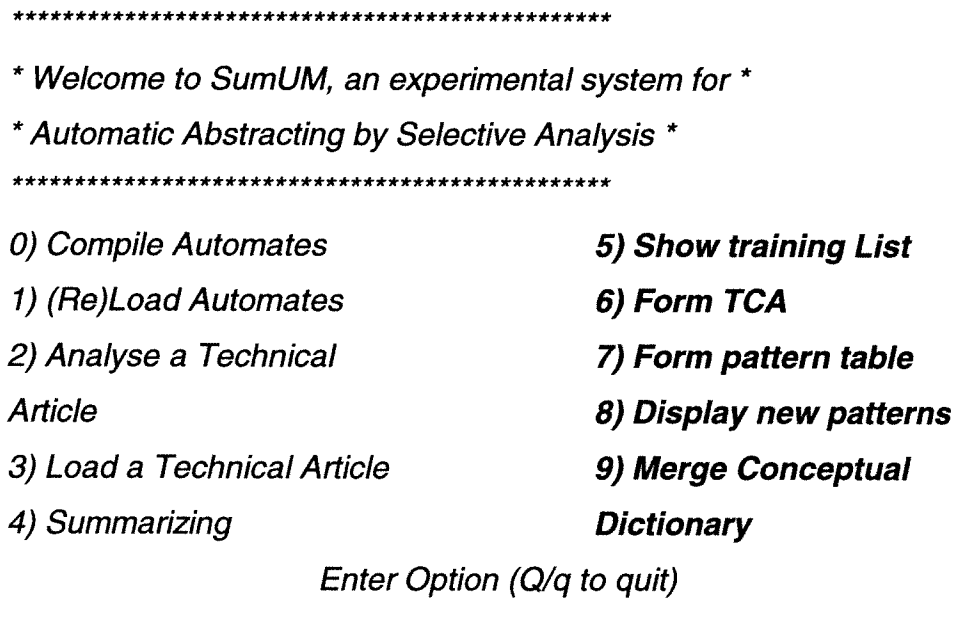


Figure 24: Interface de GlobSum et SumUM

On a débuté ce chapitre en montrant la faiblesse de ***SumUm*** sur les documents non techniques mais il est cependant impossible de faire une étude de ***GlobSum*** sur des documents non techniques, puisque premièrement ils ne sont pas structurés, deuxièmement on a pas un corpus de documents d'un autre domaine et troisièmement l'automatisation de formation de patrons n'est pas le seul facteur limitant ***SumUm*** au domaine technique (nous présenterons ces autres facteurs au **chapitre 6**). Ces pourquoi les évaluations de ***GlobSum*** ont été faites sur le corpus de documents techniques que ***SumUm*** a utilisé. Même le sommaire de cette mémoire devrait être « théoriquement » facile à produire par ***GlobSum*** si on lui fournit un corpus d'entraînement de mémoires : J'ai utilisé les guides d'écriture de mémoire, suivi le style des autres mémoires, ajouté des paragraphes liant les différentes sessions et surtout utilisé des concepts et relations dans des patrons réguliers pour présenter les sections « comme dans les autres mémoires ».

Les évaluations de ***GlobSum*** sont présentées au **chapitre 5**, les conclusions, les améliorations possibles et les travaux futures sont présentés au **chapitre 6.**

Chapitre 5

5 Évaluations de patrons et de qualité de GlobSum

Deux évaluations de ***GlobSum*** ont été effectuées, la première compare les résumés produis par ***SumUm*** aux résumés produis avec ***GlobSum***, et la deuxième compare les résumés de ***GlobSum***, par variation de la valeur du « seuil ». L'algorithme de groupage trouve tous les patrons possibles et le filtrage qui suit est basé sur ce seuil. Les formules de précisions et rappels, présentées au **chapitre 4**, sont utilisés dans nos évaluations.

L'entraînement a été effectué sur 40 des 120 documents utilisés dans la formation des patrons de ***SumUm*** correspondant à 30% du corpus. Cependant 40 documents est considéré comme « très peu » pour un entraînement convenable. Il serait intéressant de comparer ***GlobSum*** dans le cas ou un corpus plus grand (1000+ documents) serait disponible. Nous pourrions alors profiter d'un groupe d'entraînement plus élevé, soit 500+ documents. Les tests ont été effectués sur 20 documents soit 17% du corpus.

Dans la première évaluation on a pris 20 documents avec leurs résumés professionnels. On a ensuite produit les résumés de ces documents : une première fois par ***SumUm*** avec ses patrons et son dictionnaire conceptuel et une deuxième fois par ***GlobSum*** avec ses patrons et son dictionnaire conceptuel. On a ensuite calculé d'une part la précision et le rappel des résumés de ***SumUm*** aux résumés professionnels et d'autre part les résumés de ***GlobSum*** aux même résumés professionnels. Les résultats sont présentés à la **table 13**.

Numéro du Document	SumUm		GlobSum	
	Rappel	*Précision*	*Rappel*	*Précision*
1	.57	.25	.62	.20
2	.10	.11	.14	.08
3	.34	.08	.38	.08
4	.51	.06	.59	.02
5	.08	.14	.16	.07
6	.27	.23	.31	.16
7	.28	.12	.29	.08
8	.36	.19	.42	.12
9	.13	.06	.19	.06
10	.17	.09	.17	.12
11	.19	.33	.19	.14
12	.41	.38	.47	.23
13	.47	.26	.49	.21
14	.03	.23	.10	.03
15	.12	.14	.20	.10
16	.11	.13	.16	.09
17	.40	.22	.48	.18
18	.30	.24	.38	.19
19	.52	.17	.54	.12
20	.22	.19	.22	.11
Moyenne	**.28**	**.18**	**.33**	**.12**

Table 13: Rappel et précision des documents techniques par SumUm et GlobSum

La **table 13** montre une diminution de la précision de ***GlobSum*** et une augmentation du rappel. La dégradation de la précision peut être expliquer ainsi :

- L'utilisation d'un corpus et d'un groupe d'entraînement relativement petit. 120 documents au total correspondent à environ 5000 phrases, ce qui n'est pas suffisamment grand pour déterminer effectivement

quelques patrons. Dans la dernière étape de filtrage de ***GlobSum***, un seuil est utilisé pour filtrer les patrons (*le nombre de phrases qui satisfont le patron*), avec un seuil de 3%, seul les patrons satisfaits par 140 phrases et plus, sont considérés. Cette limitation du seuil peut aussi aboutir à l'élimination de patrons que ce corpus particulier n'a pas pu détecter.

- La deuxième raison de la baisse de la précision est le processus de filtrage. Dans cette étape on a utilisé le seuil et la position de la phrase pour le filtrage des groupes mais sûrement plusieurs autres facteurs pouvaient être étudiés pour parvenir à des meilleures sélections, par exemple l'ajout d'autres attributs comme le voisinage des concepts et relations ou la définition d'une valeur d'entropie et de gain d'information pour les attributs…

D'un autre coté, l'augmentation de la valeur du rappel dans ***GlobSum*** est attribuée aux causes suivantes :

- ***GlobSum*** parfois détecte des phrases non-pertinentes à cause du voisinage de concept et de relation. Par exemple une phrase de la forme suivante » « xx *ConceptA* yy *RelationB* xxxx, yyyyy xxxxx yyyy *ConceptC* » GlobSum va déterminer la cooccurrence des concepts *A,C* et la relation *B*, bien que, dans cette configuration, les concepts *A* et *C* peuvent et sont souvent indépendants à cause de leurs voisinage. Dans le cas où ***GlobSum*** a déterminé le patron de cooccurrence du concept *A* et *C* comme valide et là sauvegardé dans un template, la phrase précédente sera désormais sélectionnée juste parce qu'elle contient la même liste de concepts et de relations sans vérifier si les deux concepts A et C sont suffisamment « éloignés » pour être indépendants. Ce problème cause de mauvais remplissages de patrons et aboutit en partie à cette élévation du rappel.
- Encore une fois le seuil choisi joue un rôle dans le changement du rappel. En prenant un seuil de 3% des 5000 phrases, soit 130 phrases au moins pour satisfaire un patron, il est possible d'avoir un patron

résultant du hasard dans un corpus particulier. 130 phrases n'est pas un nombre suffisamment grand pour éliminer ce hasard.

Figure 25 et **26** présentent la variation de la précision et le rappel dans ***GlobSum*** par rapport à ***SumUm***.

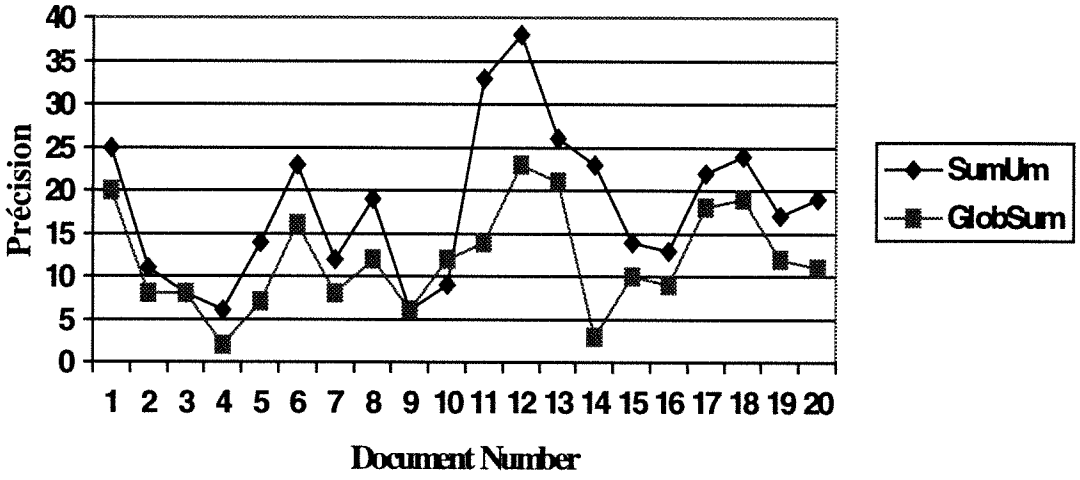


Figure 25: Variation de Précision de GlobSum

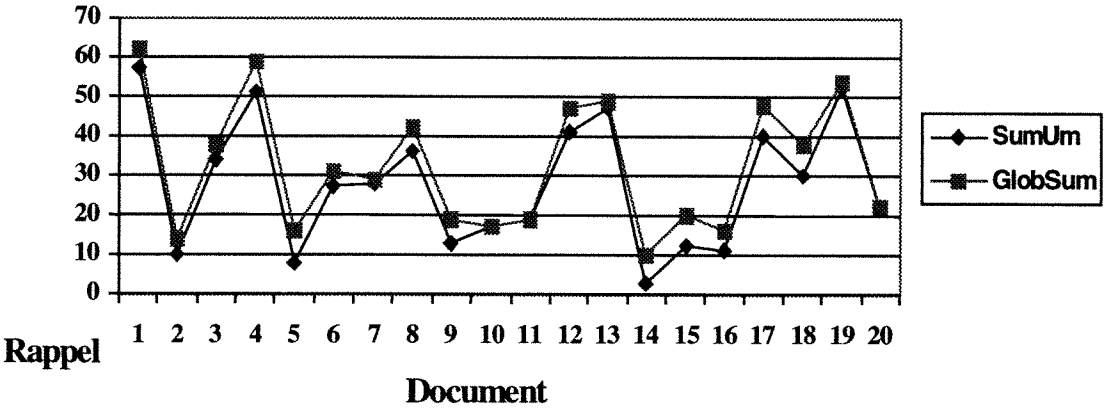


Figure 26: Variation de Rappel de GlobSum

On a vu comment la valeur du seuil peut affecter beaucoup la précision et le rappel. C'est pourquoi la seconde évaluation qu'on a effectuée compare le comportement de ***GlobSum*** par des variations du seuil. Cette évaluation sert à répondre à plusieurs questions comme : A quelle limite un groupe est il considéré comme un patron? Quel est le pourcentage de phrases suffisant pour généraliser un patron? Quel est l'effet de ces variations sur la précision et le rappel?

Trois variations du seuils ont été testées, 3%, 6% et 10% du corpus correspondant à 130, 260 et 500 phrases nécessaires pour satisfaire un patron. Les résultats et leurs effets sur la précision et le rappel sont présentés à la **table 14**.

Numéro du Document	**à 3%**		**à 6%**		**à 10%**	
	Rappel	***Précision***	***Rappel***	***Précision***	***Rappel***	***Précision***
1	.64	.22	.62	.20	.60	.22
2	.15	.06	.14	.08	.14	.08
3	.39	.08	.38	.08	.37	.08
4	.57	.03	.59	.02	.58	.04
5	.19	.05	.16	.07	.12	.10
6	.33	.13	.31	.16	.31	.15
7	.30	.08	.29	.08	.27	.08
8	.46	.11	.42	.12	.40	.14
9	.22	.06	.19	.06	.19	.08
10	.17	.11	.17	.12	.15	.12
11	.20	.12	.19	.14	.20	.15
12	.47	.20	.47	.23	.45	.24
13	.52	.19	.49	.21	.43	.21
14	.15	.01	.10	.03	.10	.04
15	.21	.07	.20	.10	.17	.11
Moyenne	**.33**	**.10**	**.31**	**.11**	**.30**	**.12**

Table 14: Changement du rappel et de la précision avec la variation du "seuil"

La **table 14** reflète l'effet du seuil. Avec une diminution à 3% correspondant dans notre corpus à 130 phrases, la précision subit une faible diminution alors que le rappel subit une faible augmentation. La diminution de la précision est causée par le nombre de phrases diminué pour acquérir un patron, plus des patrons seront identifiés et par la suite moins de phrases seront ignorées aboutissant à la sélection de phrases qui satisfont l'exigence moindre du patron, la précision alors diminue (dans la formule A augmente). D'autre part avec plus de patrons identifiés le rappel subit une augmentation (dans la formule AB subit une faible augmentation).
De l'autre coté, avec un seuil de 10%, soit 500 phrases par patron, moins de patrons sont identifiés (l'exigence de 500 documents est plus difficile à satisfaire), la précision augmente faiblement (A et AB diminuent mais la diminution de A est plus importante causant la faible augmentation de la précision) alors que le rappel subit une diminution (AB diminue) causée par l'auto filtrage des patrons satisfaits par moins de 500 phrases.

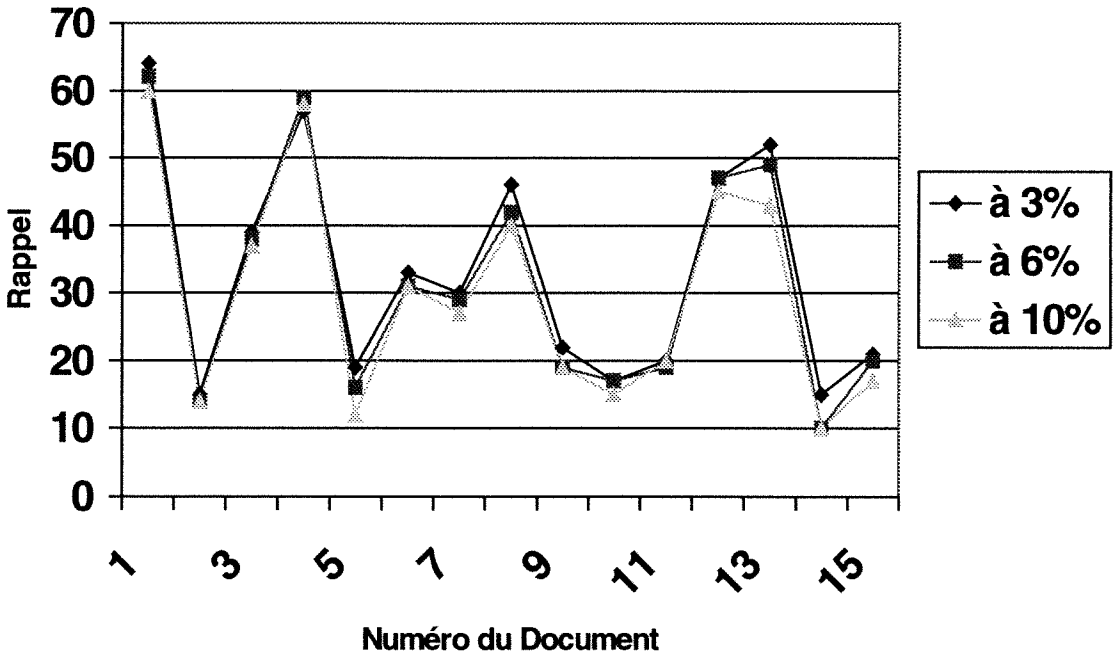


Figure 27: Variation du rappel à 3%, 6% et 10%

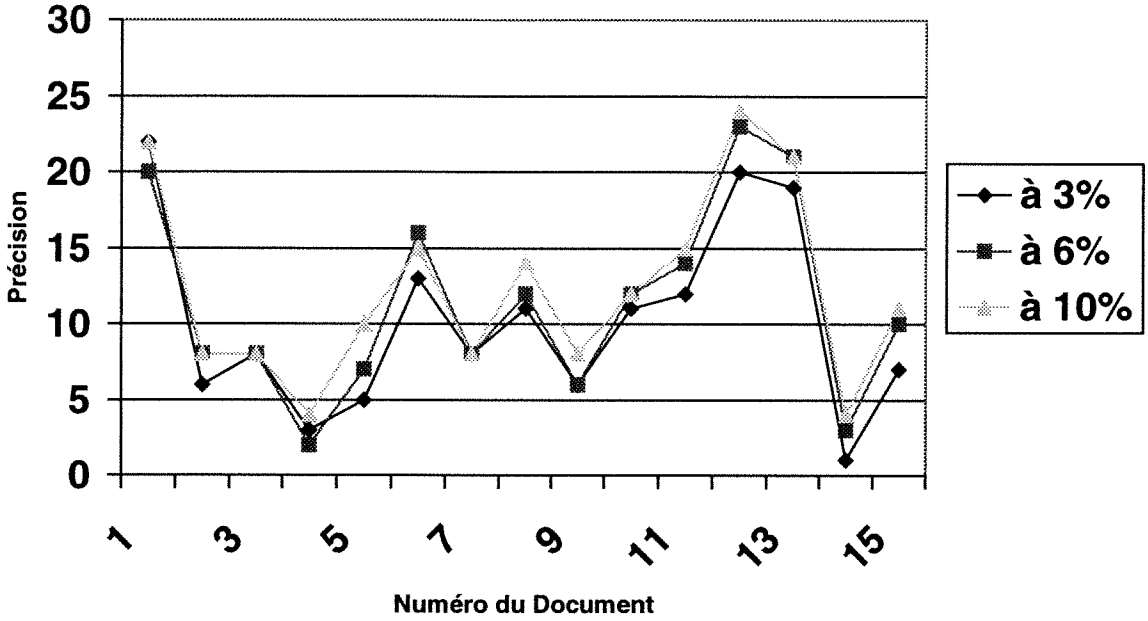


Figure 28: Variation de la précision à 3%, 6% et 10%

Les **figure 27** et **28** montrent la variation du rappel et de la précision avec le changement du « *threshold* » de 3%, 6% et 10%.

Il serait intéressant de concevoir et tester plusieurs attributs additionnels pour voir leur impact sur la précision, le rappel et la performance de *GlobSum* en général, l'attribut de voisinage entre concept et relation est un attribut capable d'améliorer la performance et la précision de ***GlobSum*** comme on l'a déjà vu. Les variables à changer dans ***GlobSum*** sont multiples et leurs différences sont intéressantes, les améliorations qu'on peut ajouter, la performance, les conclusions et les travaux futurs sont présentés dans le *chapitre 6*.

Chapitre 6

6 Conclusions et amélioration futures

La limitation majeure de ***SumUm***, un système complexe de production de résumé automatique est la restriction de domaine. Cette restriction est causée en majorité par le choix de patrons. Le but de notre système, ***GlobSum*** est d'automatiser ce processus de découverte de patrons sans changement majeur dans le comportement et la performance de ***SumUm***.

Les résultats des évaluations du **chapitre 5** montrent une détérioration de performance dans ***GlobSum.*** Cependant cette détérioration est suffisamment faible pour considérer ***GlobSum*** comme un succès partiel et les patrons dégagés par ***GlobSum*** sont semblables à ceux déterminés par appariement de corpus. Il sera intéressant alors de tester ***GlobSum*** sur un corpus non technique, une tâche qui est désormais difficile à faire puisqu'une autre limitation de ***SumUm*** et le fait qu'il s'appuie sur une structure d'introduction, conclusion, référence, auteur et section bien déterminée (*Paice C.D., 1993).*

Plusieurs possibilités d'amélioration de ***GlobSum*** sont envisageables. Les variables à tester et à modifier sont nombreuses. Dans une première étape plusieurs tests sur la valeur du seuil peuvent être faits pour déterminer la valeur « *idéale* » qui aboutit au meilleur rappel et précision, des valeurs plus dispersées peuvent être testées. La disponibilité du corpus est aussi un point à considérer. Notre corpus de 120 documents est simplement trop petit pour un entraînement idéal, les entraînements sont souvent accomplis avec un corpus de plus de 1000 documents et un groupe d'entraînement de plus de 500.

De plus, la liste d'attributs qu'on a utilisée peut être grandement améliorée par l'ajout de plusieurs types d'attribut pour améliorer les patrons. Le voisinage des concepts et relations (la distance euclidienne), la longueur de la phrase, le type de la phrase, les acronymes, les ambiguïtés...Il serait intéressant d'étudier la valeur de gain d'information (*chapitre 3, algorithme ID3*) accumulée par chaque attribut et

choisir ceux qui amènent un degré d'information maximal (*Siedlecki and Sklansky, 1988*) lors de notre filtration de groupe au lieu de la valeur simple du seuil et de la position des phrases qu'on a utilisé.

Un autre point à mentionner est l'inadaptation de ***GlobSum :*** L'algorithme découvre le style particulier pour un domaine spécifique, mais l'introduction d'un document d'un autre domaine oblige un re entraînement à ce domaine avec un nouveau corpus de document du domaine désiré. L'application de ***GlobSum*** entraîné avec un domaine *X* ne va pas marcher avec un document de domaine *Y* sans ré-entraînement.

Les chemins d'amélioration de ***GlobSum*** sont multiples : d'une part pour améliorer la performance et la complexité et d'autre part pour raffiner la précision et le rappel. Cependant avec ***GlobSum*** on a réussi à lever une des limitations majeures de ***SumUm*** afin d'arriver à un système de production de résumés automatiques puissant qui fonctionnera pour tout domaine. Deux étapes majeures restent pour supprimer totalement la dépendance de ***SumUm*** à un domaine :

1) La suppression de la contrainte de structure de ***SumUm.*** L'analyse indicative est basée sur un document qui contient des sections, titres et conclusions (normalement rencontré dans les documents techniques) et même la sélection indicative (chapitre 2, la deuxième phase de l'analyse indicative) utilise la structure topique contenant les phrases des titres, section et conclusions. Donc il est clair que ***SumUm*** se base sur la structure du document. Une solution envisageable à ce problème peut être une segmentation du document par paragraphes au lieu de sections. Tous les documents contiennent des paragraphes indépendamment de domaine et ces paragraphes peuvent remplacer les sections.
2) L'automatisation de la formation de la liste de concepts et de relations ou l'utilisation d'une liste assez grande pour tous les domaines. Dans tous nos études et tests de ***SumUm*** on a toujours supposé la disponibilité de cette liste et on a même utilisé cette liste dans ***GlobSum*** lors de l'interprétation sémantique pour dégager les concepts et relations de chaque phrase. Cette liste a été

développée par Saggion (Saggion, 2000d) après des études des documents techniques, elle est suffisamment générale pour englober un domaine technique mais elle présente des lacunes pour les autres domaines. Pour résoudre ce problème on peut essayer d'automatiser la formation de cette liste par entraînement comme on l'a fait pour les patrons : On interprète les phrases lexicalement et syntaxiquement, on forme des groupes de termes lexicaux qui sont semblable en définition et en syntaxe (on utilise un dictionnaire de termes). Si un groupe est suffisamment grand et respecte certains critères, on nomme ce groupe comme un concept/relation intitulé avec un de ces termes. Par exemple, on peut trouver un groupe contenant les termes {department, institution, school, university, college,...} et on peut, si on juge ce groupe comme suffisamment important, transformer ce groupe en concept « department » (ou un des autres termes) avec les termes lexicaux institution, school, university, college,... Bien sûr cette idée simplifiée doit contenir plusieurs méthodes d'étude de discours pour dégager de bons concepts et relations.

Même avec ***GlobSum***, plusieurs questions restent à être explorées : L'exploitation des attributs, les tests d'un plus grand corpus ou d'un corpus d'un autre domaine et les modifications de variables qui pourraient être l'objet de travaux futurs. ***SumUm*** peut sûrement arriver à offrir des résumés des documents de tous les domaines comparable et meilleur que des systèmes commerciaux.

Bibliographie

Afnor, 1984. *Recommandations aux autres auteurs des articles scientifiques et techniques pour la redaction des résumés.* Association Française de normalization.

Alterman, R. 1991. Understanding and summarization. *Encyclopedia of Artificial Intelligence*, review 5,4, 239—254.

Allen, J. F. 1994. *Natural Language Understanding*. Redwood City, CA: Benjamin/Cummings. A new edition of a classic work.

Baxter, J., 2000. *A model of inductive bias learning. Journal of Artificial Intelligence Research*, 12:149—198.

Berliner, H., and Ebeling, C. 1989. *Pattern knowledge and search: The supreme architecture*. Artificial Intelligence 38:161—198.

Boguraev B. and Kennedy C. 1997. *Salience-based content characterization of text documents*. In Proceedings of ACL'97 Workshop on Intelligent, Scalable Text Summarisation, pages 2—9, Madrid, Spain.

Breiman, Friedman, Olshen, Stone, 1984: *Classification and Decision Trees*, Wadsworth.

Charniak, E. 1993. Statistical Language Learning. Cambridge, MA: MIT Press.

Cohen P., V. Chaudhri, A. Pease, and B. Schrag, 1999. "*Does Prior Knowledge Help?*," in *Proceedings of the National Conference on Artificial Intelligence.*

Cooley, R., Mobasher, B. and Srivastava, J., 1997. "*Web Mining: Information and Pattern Discovery on the World Wide Web*; WEBMINER—a survey, "Tech Report, Department of Computer Science, University of Minnesota, Minneapolis.

Cremmins, E. 1982. *The art of abstracting*. ISI PRESS Philadelphia, PA, 150p.

Cowie and W. Lehnert, 1996. "*Information Extraction*," Comm. ACM, Vol. 39, No.1, pp. 80--91.

DeJong, G., Mooney, R., 1986. *Explanation based learning: An alternative view*, Machine Learning, 1(2):145-176.

Edmundson H.P., 1969, "*New methods in automatic extracting*" in Journal of the ACM, 16(2): 264-285.

Ernst, G. W. and Newell, A. 1969. *GPS: A Case study in generality and problem solving*. New York: Academic Press.

Everitt, Brian, 1980. *Cluster analysis*. Halsted Press, New York.

Feigunbaum, E. A., 1963. *The simulation of verbal learning behavior*. In Feiginbaum and Feldman (1963).

Freeman, H., Ahn, J., 1987. *"On the problem of placing Names in a Geographic Map,"* Int'l J. on Patt. Recog. & AI, pp. 121-140.

Fikes. R. E. and Nilsson, N. J. 1971. *STRIPS: A new approach to the application of theorem proving to artificial intelligence*. Artificial Intelligence, 1(2).

Grossberg, S. 1976. Adaptive pattern classification and universal recoding: 1. parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23:121 – 134

Grossman John, 1993. The Chicago Manual of Style : *The Essential Guide for Writers, Editors, and Publishers* (14th Edition).

Holland, J. H. 1986. *Escaping brittleness : The possibilities of general purpose learning algorithms applied to parallel rule-based systems*. In Michalski et al.

Jan L. Talmon, 1986. *A multiclass non parametric partitioning algorithm*. Pattern Recognition Letters, 4L31#38.

Jin-Young Ha, Se-Chang Oh and Jin H. Kim, 1995. *« Recognition of Unconstrained Handwritten English Words with Caracter and Ligature Modeling* », Int. Journal of Pattern Recognition and Artificial Intelligence, 9, 3 pp 535-556.

Kameyama M. and Arima I., 1994. « *Coping with aboutness complexity in information extraction from spoken dialogues.* » in the proceedings of the international conference on spoken language processing (ICSLP-94), (Yokohama, Japan)

Kohonen, 1995. Teuvo. Self-organizing maps. Berlin; Heidelberg; New-York: Springer. ISBN: 3-540-58600-8.

Lenat, 1977.The automated Mathematicien, AM Thesis statement, University of Pennsylvania.

Lengagne R.. Fua, and O. Monga, 1996. *Using Crest Line to Guide Surface Reconstruction from Stereo.*" in *International Conference on Pattern Recognition*, (Lausanne, Switzerland).

Lin C. and Hovy, E. 1997. *Identifying Topics by Position.* In fifth conference on applied natural language processing, pages 283-290. Association for computational linguistics

Maizell R.E. and Smith J.F. and Singer T.E.R., 1971. "*Abstracting Scientific and Technical Literature: An Introductory Guide and Text for Scientists, Abstractors, and Management*", in Wiley-Interscience, New York.

Maybury, M. 1995. *Generating summaries from event data.* Information Processing & Management 31, 5, 735--751.

McKeown, K., and W. Swartout. 1987. *Language Generation and Explanation.* In Annual Review of Computer Science, Vol. 2, Palo Alto, CA: Annual Reviews, p 1-51.

Mitchell, T.M, 1977, *Version Spaces: A candidate Elimination approach to rule learning*. Proceedings fifth international joint conference on artificial intelligence, Cambridge, Mass. Pp 305-310.

Mitchell T. M. 1980. *The need for the biases in learning generalization.* Technical Report CBM-TR-177, Department of Computer Science, Rutgers University, New Brunswick, NJ.

Mitchell T, M., 1982. *Generalization as search.* Artificial intelligence, 18(2): 203-226.

Mitchell Tom M., McGraw-Hill, 1997. *Machine Learning, chap11.*

Moore R.C., 1994. *Integration of speech with natural language understanding.* In D.B Roe and J.G Wilpon, editors, Voice communication between human and Machines, pages 254—271. National Academy Press, Washington D.C., USA.

Moore R.C., 1998. "*Using natural language knowledge sources in speech recognition,*" in *Proceedings of the NATO Advanced Study Institute (ASI)*, NATO.

Paice, C., 1990. *Constructing literature Abstracts by computer*: Techniques and Prospects., Information Processing & Management, 26(1): 171-186.

Paice C.D., 1993. "*The Identification of Important Concepts in Highly Structured Technical Papers*" in Proceedings of the Sixteenth Annual International ACM SIGIR conference on Research \& Development in IR.

Quinlan, J.R., 1986. *Induction of decision trees*. Machine Learning, 1(1):81-106.

Quinlan, J.R., 1987. *Simplifying decision trees*, international Journal of Man-Machine Studies

Reitman, W.R. 1965. *Cognition and though*. New York: Wiley.

Reitman, J.S., Holland, J., 1978. *cognitive systems based on adaptive systems*, Pattern directed inference systems, pp. 313-329, ED. D.A Waterman et al., 1978.

Rijsbergen, C., and Williams, P., 1979. editors, *Information Retrieval Research*, London: Butterworth, p 118-130.

Riloff E., 1993. "*A Corpus-Based Approach to Domain-Specific Text Summarisation: A Proposal*", in B. Endres-Niggemeyer, J. Hobbs, and K. Sparck Jones editions, Workshop on Summarising Text for Intelligent Communication.

Riloff E., 1996. *Automatically Generating Extraction Pattern from Untagged Text*, Proc. Of American association for artificial intelligence (AAAI-96), pp. 1044-1049.

Rosenblatt, F. 1958. *The perceptron: A probabilistic model for information storage and organization in the brain*. Psycological Review, 65:386-408

Ross, P. 1989. *Advanced Prolog*. Reading, MA: Addison-Wesley.

Saggion, H., Lapalme, G., 1998. *Where does information come from*? Corpus Analysis for Automatic Abstracting. In Rencontre Internationale sur l'extraction de filtrage et le résumé automatique. RIFRA'98, pages 72-83, Sfax, Tunisie.

Saggion, H., Lapalme, G., 2000a. *Selective Analysis for the automatic Generation of Summaries*. In proceedings of the 6th international conference of the international society for knowledge organization, Faculty of information studies. University of Toronto. Toronto, Ontario, Canada. Salton-1 G., 1991. Development in automatic text retrieval. Science, 253:974—980.

Saggion, H., Lapalme, G., 2000b. *Concept identification and presentation in the context of technical text summarization*. In proceedings of the workshop on automatic summarization, Seatle, USA. Association for computational linguistics.

Saggion, H., Lapalme, G., 2000c. *Evaluation of content and text quality in the context of technical text summarization*. In proceedings of the computer-assisted information searching on internet conference. RIAO'2000, Paris, France.

Saggion, H., 2000d. Generation automatique de résumé par analyse selective. Departement d'information et de recherche operationnelle, faculté des arts et des sciences. These de PhD.

Salton G. and Mitra M. and Buckley C. and Singha A., 1994. "*Automatic Text Generation and the Analysis of Text Structure*".

Salton G., Y. Chiaramella, X., A. Bookstein, 1991. "A self organizing semantic map for information retrieval", SIGIR'91 – Proc. Fourteenth Annual International ACM/SIGIR Conference on research and development in information retrieval, Eds. Lin, D. ACM Press, pp 262—269

Scoot B., 1995. Huffman. Learning information extraction patterns from examples in IJCAI-95 Workshop on New Approaches to learning for NLP.

Shannon C., 1948. A mathematical theory of communication. Bell system Technical Journal

Shavlik J. W., T G Dietterich, 1990. Readings in Machine Learning. Morgan Kaufmann. pp 20-55.

SICStus 1998, *SICStus Prolog User's Manual*. The Intelligent Systems Laboratory. Swedish Institute of Computer Science.

Siedlecki, W. & Sklansky, J. 1988. "*On automatic feature selection*", International Journal of pattern recognition and artificial intelligence 2(2), pp. 197—220.

Tait J.I., 1983. "*Automatic summarizing of English texts*". Technical Report No. 47, University of Cambridge Computer Laboratory.

Therrien, C. W., 1989. Decision Estimation and Classification: An introduction to pattern recognition and related topics. John Wiley and Sons.

Teufel S., and Moens M., 1997. Sentence extraction as a classification task. In ACL/EACL-97 workshop on intelligent Scalable text summarization. Madrid, Spain, pp. 58—65.

Thomas J., Le Roux B, 1996. GDM refinements as learning Bias refinements. *EKAW'96 (Position Papers), Nottingham, UK.*

Turney, P. 1999. Learning to Extract Keyphrases from Text. Technical Report NRC Technical Report ERB-1051, National Research Council of Canada.

Utgoff, P. E., 1986. Shift of bias in inductive concept learning. In Michalski et al.

Walker, A., McCord, M., Sowa, J.F., and Wilson, W. G. 1987. *Knowledge Systems and PROLOG: A logical approach to expert systems and natural language processing.* Reading, MA: Addison-Wesley.

Walczak, S. 1992. Pattern-based tactical planning. International Journal of Pattern Recognition and Artificial intelligence, 6(5), 955-988.

Annexe A
Liste de Concepts

Les concepts et les relations sont ceux utilisés par *Horacio Saggion* pendant le développement de ***SumUm***. Les termes lexicaux (Lexical Items) attribués à chaque concepts et relation sont présentés en partie seulement.

Concepts	**Explanation & Example**	**Lexical Items**
Objective	The general objectives, "...the natural focus is on the third of the *above goals...*"	***Goal, objectives, ...***
Description	A description. "...the *description* of a problem, that has been..."	***Definition, ...***
Analysis	An analysis. "To enable such *an analysis* of problem solving	***Analysis, ...***
Conclusion	Conclusion of the paper. "*our conclusion* was that simple and local transformations..."	***Conclusion, ...***
Discovery	A discovery. "...to guide *the discovery* of repetitive functional substructures in large structural databases."	***Discovery,...***
Title	A title from the document. "2.3. *Algorithm Implementation*"	***Number Title, ...***
Solution	The solution to the problem. "In this paper, we describe the problem and propose *a solution*, ..."	***Solution, answer, ...***
Study	The object of study. "an empirical *study* of randomly generated binary..."	***Study, ...***
Author	The authors of the article. "*I* refer to ..."	***We, I, author, ...***
Survey	A survey. "...Ameritech Library Services collaborated on a *survey* of electronic services"	***Survey, ...***
Example	An example. "*One example* is "concept formation" as a goal, ..."	***Example, illustration, ...***
Focus	The general focus. "A key *focus* of the technical specification was ..."	***Focus, ...***
Comparison	A comparison. "Aamodt's *comparison* of knowledge intensive CBR methods	***Comparison, ...***
Date	A date. "*In 1938* Albert Einstein and Leopold Infeld wrote..."	***Sequence of digits, ...***
Problem	The problem under consideration. "the *lack of* a library severely limits the impact of..."	***Difficulty, issue, problem, ...***
Work	The work of the author. "*The work* described in this paper addresses..."	***Work, ...***
Researcher	Other researchers. "*Cannon* shown..."	***Proper Noun...***
Project	A (research) project. "*A research project* currently in progress at..."	***Project, ...***
Acronym	An acronym. "The world wide web (*WWW*)..."	**Noun Group(*Acronym*), ...**
Acronym expansion	The expansion of the acronym. "*The world Wide Web* (WWW)..."	***Noun Group(Acronym), ...***
Conceptual Objective	The objective of a domain concept. "*The aim of the* DECIMAL Project is to..."	***Goal of conceptual entity, ...***
Explanation	An explanation. "...and *an explanation* of ground facts cannot lead to rules."	***Explanation, ...***
Paper component	A component of the research paper. "...some successful applications (*Section 3*)..."	***Section, subsection, ...***
Need	A necessity.	***Need, necessity, ...***

	"...*the need* for an interface between..."	
Research paper	A research group. "...is *a staff programmer* in the experimental Systems group at the ..."	***Article, here, paper, ...***
Institutions	Institutions. "*Department* of mechanical and..."	***University, ...***
Method	The method used in the study "One approach is to support indexing by the traditional *method of...*"	***Equipment, methodology, ...***
Topic	The general topic. "*These main topics* are natural-language processing..."	***Topic, theme, ...***
Presentation	A presentation. "...*a logical definition of* the abductive problem..."	***Definition, ...***
Summary	Summary of information. "...*a summary of* the fundings of the research phase."	***Summary, ...***
Suggestion	A suggestion. "*A number of suggestions* are put in place for..."	***Suggestion, ...***
Results	The results obtained. "*The results* indicate that..."	***Result, outcome, ...***
Quantity	A quantity. "...capable of integrating voice, data (*64 Kb/s to 2 MB/s*) ..."	***Number, ...***
Experiment	The experiment. "*The experiments* were done in order ..."	***Experiment, test, ...***
Affiliation	The affiliation of the authors. Ananth Y. Grama, *Pursue* University	***Prop.Noun, Institution, ...***
Others' paper	The article of other researchers. "*In their article...*"	***Article, ...***
Situation	The situation. "*the Austrian situation* in the field of telecommunication is far behind	***Situation, today, ...***
Advantage	Advantage. "...*the advantages of* CBR..."	***Advantage, asset, ...***
Description	A description. "...*the description of* a problem, that has been..."	***Description, ...***
Development	A development. "...the AI approach has prevailed in the *implementation* of high-level planning"	***Development, ...***
Question	Research question. "...to understand *how services* which are of value.."	***Research question, ...***
Mathematical	Mathematical entity. "*Polynomial equations* are used for representing..."	***Formula, equation, ...***
Research	The research work. "...a broad range of *scientific research...*"	***Research, ...***
Author related	Authors related entity. "The core of *our system* is comprised of..."	***Our, my, ...***
Research group	A research group. "...is a staff programmer in *the experiment systems* group at the..."	***Group, ...***
Captioning	The captioning of an element. "*Fig.2:* Test Results."	***Figure X Captioning, ...***
Structural	Structural element of the document such as a table or figure. "*In figure 3* we show..."	***Figure, table, picture, ...***
Reference	Reference to previous work. "...the Manhattan street network [*4,5*]."	***Proper Noun(Year), ...***
Application	Applicability. "...*the applicability* of our approach..."	***Use, employ, ...***
Discussion	A discussion. "*our discussion* focuses on the main..."	***Discussion, ...***
Conceptual focus	The focus of a concept. "*The focus* of the paper is..."	***Focus of conceptual entity, ...***
Overview	An overview. "... give *a brief overview* of the AI.."	***Overview, ...***
Conceptual topic	The topic of the paper. "...a particularly important *topic of study...*"	***Topic of article, ...***
Introduction	Introducing information. "*a brief introduction* to this..."	***Introduction, ...***
Hypothesis	The hypothesis. "that is, in other words, to get the most *reasonable hypothesis.*"	***Hypothesis, assumption, ...***

Annexe B
Liste de Relations

Les concepts et les relations sont ceux utilisés par *Horacio Saggion* pendant le développement de ***SumUm***. Les termes lexicaux (Lexical Items) attribués à chaque concepts et relation sont présentés en partie seulement.

Relations	**Explanation & Example**	**Lexical Items**
Discover	Discovering. "Significant reduction...*were discovered...*"	***Determine, discover, ...***
Close	Closing. "The paper *concludes with* observations on the potential..."	***Conclude, close, ...***
Focus	Identifying the objective. "*Its charter is* to perform research and development in advanced information technology..."	***Focus on, ...***
Define	Defining. "Azuma (1997) *has defined* augmented reality system as..."	***Define, to be, ...***
elaborate	Elaborating. "This properties *allow us to* use MT to express and prove tactics"	***Allow, contribute, ...***
Advantage	Identifying advantage. "...simulated annealing and evolutionary programming *outperform back...*	***To have advantage, ...***
Need	Identifying need. "In order to understand the French situation it *is necessary to* describe..."	***To be a necessity, ...***
Open	Opening. "*Starts from* a point-based metric system and gives a construction of ..."	***Begin, start, ...***
Comment	Commenting. "Notes the *continuing* explosion *of* information..."	***Mention, note, ...***
Problem	Identifying. Problem. "...the main control problems that *arise* when ..."	***Arise, complicate, ...***
Practical	Identifying practicality. "V&V methodologies is *a practical ...*"	***To be practical, ...***
Use	Identifying usefulness. "The analytical too which *is used for* this purpose ..."	***Apply, employ, ...***
Conclude	Concluding. "The second conclusion *to be drawn is* that..."	***Conclude, ...***
Describe	Describing. "The classical generative planning process *consists of* a search..."	***Compose, form, ...***
Infer	Inferring. "The combination of the two methods...*has been proven...*"	***Prove, infer, ...***
Make known	Introducing the topic of the paper. "In this paper *we present...*"	***Describe, expose, ...***
Create	Bringing into being. "A new generation of systems *are being developed ...*"	***Build, complete, ...***
Relevance	Identifying relevance. "Combinatorial and geometric computing *is a core* area..."	***To be central, ...***
Solution	Identifying solution. "...it *overcomes* many of the past barriers to..."	***Overcome, solve, ...***
Perform	Doing. "..genetic programming optimized for *performing...*"	***Perform, ...***
Essential	Identifying essentiality. "It *is essential* that all information staff..."	***To be essential, ...***

Show	Identifying graphical material. "Figure 1 *illustrates* the concept…"	***See, show, …***
Experiment	To do experiments. "*Experiments* were done in which…"	***Do experiment, …***
Novel	Identifying novelty. "The possibility of *this new* computing paradigm…"	***To be new, …***
Identify	Characterizing entity. "…a new algorithm *called* OPT-2 for optimal…"	***Contain, classify, call, …***
Investigate	Investigating. " The phrase transition in binary, i.e.… *is investigated.*	***Investigate, …***
Summarize	Summarizing. "This analysis *summarizes* some of the work…"	***Sum up, …***
Evidence	Giving evidence. "*Evidence* for the applicability *is provided, …*"	***To be evident, …***
Interest	Express interest. "We *are concerned with* production of…"	***Address, concern, …***
Argue	Argumenting. "It *is argued* that for most uses which are…"	***Argue, give argument, …***
Exemplify	Exemplifying. "Other *examples* of robotic systems are also presented…"	***To be example, …***
Positive	Identifying positiveness. "…the workstations *are promising"*	***To be positive, …***
Study	Studying a topic. "The possible reasons for this are *explored* as well as…"	***Analyse, examine, …***
Situation	Identifying the situation. "Genetic algorithms, the best *known* of the variants.	***Known, …***
Opinion	Making a judgement. "A class of problems *is considered…*"	***Consider, believe, …***
Objective	Identifying the objective. "*Its charter is* to perform research in advanced information…"	***Aim, …***
Effective	Identifying effectiveness. "Our algorithm is *effective for…*"	***To be effective, …***
Explain	Explaining. "The accuracy of a prediction based on the *number is discussed…*"	***Discuss, explain, …***

Annexe C
Liste des Journaux des documents sources

1. *AI Communications*
2. *AI Magazine*
3. *American Libraries*
4. *Annals of Library Science & Documentation*
5. *Archives and Museum Informatics*
6. *Artificial Intelligence*
7. *Aslib Proceedings*
8. *Australian Library Journal*
9. *Bottom Line*
10. *Byte*
11. *Collection Management*
12. *College & Research Libraries*
13. *Computer Communications*
14. *Computer Networks and ISDN Systems*
15. *Computers in Libraries*
16. *Digital Publishing Technologies*
17. *Document Delivery & Information Supply*
18. *Electronic Library*
19. *Electronic Publishing*
20. *IATUL Proceedings*
21. *IEEE Expert*
22. *Information Outlook*
23. *International Journal of Human-Computer Studies*
24. *Interacting with Computers*
25. *Journal of End User Computing*
26. *Journal of Government Information*
27. *Journal of Interlibrary Loan*
28. *Journal of Library Administration & Management*
29. *Library Association Record*
30. *Library Hi Tech News*
31. *Library Journal*
32. *Libri*
33. *Microform & Imaging Review*
34. *New Review of Hypermedia and Multimedia*
35. *New Review of Information & Library Research*
36. *New Review of Information Science & Library Research*
37. *OCLC Newsletter*
38. *Scandinavian Public Library*
39. *Telematics and Informatics*
40. *Vine*

Annexe D
Liste des Documents non techniques

1. **Clinton's eight reasons for Rich pardon:** Former President Clinton cited eight reasons for his pardons of financiers Marc Rich in Sunday's New York Times.
   http://www.cnn.com/2001/ALLPOLITICS/02/18/clinton.fact.check/index.html - February 18, 2001

2. **Nortel CEO sees rebound:** Hard hit by an abrupt downturn in the U.S. economy, Nortel Networks Corp.'s chief executive said that all of its customers are adjusting their budgets for the slowdown. *http://cnnfn.cnn.com/2001/02/19/technology/nortel/index.htm - February 19, 2001*

3. **Hall of Famer Eddie Mathews:** Eddie Mathews died at age 69 after a long illness. He had been hospitalized with heart problems. Mathews was the only person to play for the Braves in Boston, Milwaukee and Atlanta. *http://sportsillustrated.cnn.com/baseball/mlb/news/2001/02/18/t1_mathews/index.html - February 19, 2001*

4. **Angry farmers launch cull protest:** Germany, more than 1,000 farmers gathered in Dresden in protest at the government's mass slaughter of cattle. ***http://www.cnn.com/2001/WORLD/europe/germany/02/17/farmer.protest/index.html - February 17, 2001***

5. **KFOR action**: Yugoslavia demands peacekeepers in the tense buffer zone between Serbia and Kosovo take a tougher line against Albanian extremists. *http://www.cnn.com/2001/WORLD/europe/02/19/kosovo.03/index.html - February 19, 2001*

6. **More than 1 in 4 U.S. bridges are deficient:** Surveys showed that one out of every four bridges in the U.S. isn't strong enough to support the traffic it carries.
   http://cnn.com/2001/LOCAL/westcentral/02/22/KMBC.bridges/index.html - February 22, 2001

7. **Venezuelan voodoo alive and kicking:** While Venezuela grows increasingly poor, more and more people turn to country's dense forests calling upon the help of witches, magicians and higher spirits. *http://www.thisiscyberia.com/environment/venezuela.asp - February 19, 2001*

8. **Green Bits, Turkish imam prays for snow:** The spiritual leader of Muratgoren village in the mountains of southeastern Turkey held an outdoor

service, urging villagers to pray for snow, as an increasingly bad drought threatens this year's harvest.
http://www.thisiscyberia.com/environment/news.asp - February 19, 2001

Annexe E
Liste des documents du Corpus de Test (25 au total)

Les documents font partie des journaux cités à **l'annexe C** en format : ***Titre du Document***, ***Nom du Fichier du Document***. Les fichiers sont sauvegardés dans le répertoire « ***/u/tout/Tests*** »

1. *Importing and reshaping digitized data for use in rapid prototyping: a system for sculpting polygonal mesh surfaces,* **Alciatore**
2. *Image indexing and retrieval: some problems and proposed solutions,* **Baxter**
3. *"Soft" grasping using a dextrous hand,* **Caldwell**
4. *Climbing, walking and intervention robots,* **Developpement**
5. *Strategies for content migration on the World Wide Web,* **Evans**
6. *Climbing, walking and intervention robots,* **Focus**
7. *An overview of catalog design problems in resource discovery,* **Goodchild**
8. *Enhancing the quality of low bit-rate real-time Internet communication services,* **Hui**
9. *Surface macro-texture design for rapid prototyping,* **Jee**
10. *A distributed component framework for integrated network and systems management,* **Knahl**
11. *RobotScript: the introduction of a universal robot programming language,* **Lapham**
12. *Class II MHC quantitative binding motifs derived from a large molecular database with a versatile iterative stepwise discriminant analysis meta-algorithm,* **Mallios**
13. *The NASA Technical Report Server,* **Nelson**
14. *D. J. Fonseca and G. M. Knapp Department of Industrial Engineering, The University of Alabama,* **Otro**
15. *Cobots,* **Peshkin**
16. *The Amadeus project: an overview,* **Robinson**
17. *Adaptive slicing using stepwise uniform refinement,* **Sabourin**

18. *A mobile robot for pressurizer inspection,* **Sun**

19. *Telexistence and R-Cubed,* **Tachi**

20. *The McKibben muscle and its use in actuating robot-arms showing similarities with human arm behaviour,* **Tondu**

21. *Internet conferencing with networked virtual environments,* **Towell**

22. *Application of suspension mechanisms for low powered robot tasks,* **Uddin**

23. *Climbing, walking and intervention robots,* **Validacion**

24. *Designing for human-robot symbiosis,* **Wilkes**

25. *Conceptual framework for the thermal process modelling of fused deposition,* **Yardimci**

Annexe F
Liste des documents d'entraînement (40 au total)

Les documents font partie des journaux cités à **l'annexe C** en format : ***Titre du Document***, ***Nom du Fichier du Document***. Les fichiers sont sauvegardés dans le répertoire « ***/u/tout/Tests*** »

1. *A top-down methodology for building corporate Web applications,* **Artz**
2. *Robotic system for collaborative control in minimally invasive surgery,* **Bernard**
3. *A client-side Web agent for document categorization,* **Boley**
4. *Automated pipe inspection robot,* **Bright**
5. *Design and implementation of an aided fruit-harvesting robot (Agribot),* **Ceres**
6. *Application Performance on the MIT Alewife Machine,* **Chong**
7. *Features 3D scanning systems for rapid prototyping,* **Clark**
8. *A framework for evaluating Internet telephony systems,* **Foo**
9. *A learning organization perspective on training: Critical success factors for Internet implementation,* **Hert**
10. *Surface macro-texture design for rapid prototyping,* **Jee**
11. *Vision and force/torque sensing for calibration of industrial robots,* **Lin**
12. *A knowledge-based approach to domain-specialized information agents,* **Loke**
13. *Transparent telepresence research,* **Mais**
14. *Cognitive maps, AI agents and personalized virtual environments in Internet learning experiences,* **Maule**
15. *InfoMall: an innovative strategy for high-performance computing and communications applications development,* **Mills**
16. *Local sequence alignments with monotonic gap penalties,* **Mott**
17. *A conjoint model for Internet shopping malls using customer's purchasing data,* **Otro1**
18. *Genetic algorithm based defect identification system,* **Otro5**

19. *Characterization of the laminated object manufacturing (LOM) process,* **Park**
20. *Advances in microwave MCM-D technology,* **Pieters**
21. *Software piracy among academics: an empirical study in Brunei Darussalam,* **Rahim**
22. *Vision again the star at Manufacturing Week,* **Rooks**
23. *Packaging of closed chamber PCR-chips for DNA amplification,* **Schab**
24. *Evaluating domestic and international Web-site strategies,* **Simeon**
25. *Climbing, walking and intervention robots,* **Study**
26. *Using change-point detection to support artificial neural networks for interest rates forecasting,* **Otro9**
27. *The KRAFT architecture for knowledge fusion and transformation,* **Otro13**
28. *Impact of ATM switch architectures on CBR video performance,* **Otro15**
29. *Adrenomedullin augments nitric oxide and tetrahydrobioptein synthesis in cytokine-stimulated vascular smooth muscle cells,* **Otro18**
30. *Control of a fluid catalytic cracking unit based on proportional-integral reduced order observers,* **Otro20**
31. *Microbial protein supply from the rumen,* **Otro22**
32. *Separation of glucose/fructose mixtures: counter-current adsorption system,* **Otro25**
33. *Issues and experimental results in vision-guided robotic grasping of static or moving objects,* **Papani**
34. *Type of Article: Case study, Technical,* **Towel**
35. *Application of suspension mechanisms for low powered robot tasks,* **Uddin**
36. *Intranet document management systems,* **Wen**
37. *Designing for human-robot symbiosis,* **Wilkes1**
38. *Teleoperator slave - WAM design methodology,* **Townsend**
39. *Towards the integrated monitoring and evaluation system IMES: a real power,* **Psarras**
40. *Mechanical properties of short-fibre layered composites: prediction and experiment,* **Zak**