MASTER EN SCIENCES INFORMATIQUES

Consultation des bases de données en langage naturel pour le logiciel EXPESURF

Popa, Irina-Georgiana

Award date:
2011

Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'Informatique
Année Académique 2010 – 2011

Consultation des bases de données en langage naturel pour le logiciel EXPESURF

Étudiante : Irina-Georgiana Popa

Mémoire présenté en vue de l'obtention du grade de
master en Sciences Informatiques

VTLS 20206386

RÉSUMÉ

Tim Menzies a dit « *Je rêve encore au jour où mon processeur WORD écrirait un article comme celui-ci, pendant que moi je vais à la plage !* » Y arrivera-t-on un jour ? On n'est pas sûr, mais pour atteindre cet objectif le traitement automatique du langage naturel, qui aborde la compréhension de la langue naturelle réalisée par ordinateur, est une étape essentielle.

Ce mémoire présente le problème d'interrogation en langage naturel des bases de données, une application directe du domaine de traitement automatique du langage naturel. Le but est de transformer une question en langue française en une requête d'interrogation de la base de données. L'avantage principal de l'utilisation du langage naturel est que cela ne demande pas à l'utilisateur de posséder des connaissances de la structure de base de données. La solution que nous proposons s'oriente vers l'analyse syntaxique de la phrase interrogative pour une partie restreinte de la langue française. Elle s'appuie sur la constitution d'un dictionnaire avec des mots clés et d'une bibliothèque des opérateurs pour le domaine d'application du traitement de la surface des matériaux, en particulier pour le logiciel du système expert EXPESURF.

Mots clés : langage naturel, traitement automatique du langage naturel, interrogation de la base de données, système de question-réponse, interrogative sentence, opérateur linguistique, système expert, EXPESURF

ABSTRACT

Tim Menzies said "I still dream of the day when my WORD processor writes an article like this, while I go to the beach. Will that day arrive?" We are not sure, but in order to achieve this objective the natural language processing, that tackles the understanding of the natural language realized by the computer, is an essential step.

This dissertation presents the problems faced when interrogating a database using natural language, an application which is part of the natural language processing domain. The purpose of this research is to parse a question in French language in to a database query. The main advantage of using the natural language is that the user does not require any previous knowledge of the database structure. The solution we are proposing is towards a syntactic analysis of the interrogative sentence for a restraint part of the French language. It is based on the generation of libraries with keywords and linguistics operators for the material surface treatment application domain, in particular for the EXPESURF expert system software.

Keywords: natural language, natural language processing, database querying, question - answer system, interrogative sentence, linguistics operators, expert system, EXPESURF

REMERCIEMENT

J'aimerais remercier ici chaleureusement certaines personnes qui m'ont particulièrement soutenue dans ce travail.

Tout d'abord, je désire remercier sincèrement mon promoteur, le Professeur Jean-Marie Jacquet, pour sa patience et sa disponibilité et aussi pour les connaissances transmises.

Merci sincèrement à Isabelle Linden pour ses conseils pratiques, ses suggestions et ses corrections sur mon travail.

Un grand merci particulièrement à Octavian pour sa compréhension, son support moral, ses conseils et son encouragement pendant les moments les plus difficiles.

Merci également à ma famille et mes amis qui m'ont soutenue et encouragée à poursuivre ces études.

À tous, encore une fois, un grand merci !

TABLES DE MATIÈRES

RÉSUMÉ 3

REMERCIEMENT 4

1. INTRODUCTION 7

1.1. Délimitation du sujet 8
1.1.1.Traitement Automatique du Langage Naturel 8
1.1.2. Pensée humaine versus intelligence artificielle 9
1.1.3. Les systèmes de questions - réponse 10

1.2. Le logiciel EXPESURF 11

1.3. L'objectif de l'étude 12

1.4. Relations avec autres travaux 13

1.5. Structure du mémoire 13

2. SYSTEME DE QUESTION – REPONSE 15

2.1. Introduction 15

2.2. Intelligence artificielle 16

2.3. Traitement automatique du langage naturel (TALN) 20
2.3.1. Les étapes du TALN 21
2.3.2. Applications de TALN 25

2.4. Système de question-réponse 27
2.4.1. Histoire 27
2.4.2. Interaction homme-machine 29
2.4.3. Recherche d'informations 29

2.5. Conclusion 31

3. DEVELOPPEMENT D'UN SYSTEME DE QUESTION-REPONSE 33

3.1. Introduction 33

3.2. Architecture 34
3.2.1. Analyse de question 35
3.2.2. Recherche des documents 36
3.2.3. Extraction des réponses 37

3.3. Difficultés rencontrées 41

3.4. Évaluation 42

3.5. Outils pour les systèmes de question-réponse 45
3.5.1. Outils de base des données 45
3.5.2. Outils de développement 48

3.5.3. Outils de développement d'interface 53
3.5.4. Outils de traitement automatique du langage naturel 55
3.5.5. Le système de question-réponse pour le logiciel EXPESURF et les outils utilisés 57

3.6. Conclusion **57**

4. ÉTUDE DE CAS APPLIQUE AU LOGICIEL EXPESURF **59**

4.1. Introduction **59**

4.2. EXPESURF **59**
4.2.1. La structure du logiciel EXPESURF 59
4.2.2. La structure de la base de données 62
4.2.3. Pourquoi un système de question-réponse ? 70

4.3. Grammaire : la phrase interrogative **71**
4.3.1. Type de la phrase interrogative 71
4.3.2. Structure de la phrase interrogative directe 72
4.3.3. Mots interrogatifs de la proposition interrogative 73

4.4. Système de question-réponse pour le logiciel EXPESURF **75**
4.4.1. Présentation générale 75
4.4.2. La structure du système 76
4.4.3. Les outils créés pour la consultation de base de données en langage naturel 77
4.4.4. Le dictionnaire 83
4.4.5. Les opérateurs 84
4.4.6. Compléments sur l'exemple 87
4.4.7. Limitations 88
4.4.8. Portabilité 89
4.4.9. Documentation 91

4.5. Conclusion **91**

5. CONCLUSION **93**

BIBLIOGRAPHIE **95**

GLOSSAIRE DES ACRONYMES **99**

ANNEXES **101**

Annexe A : Liste des figures **101**

Annexe B : Liste des équations **102**

Annexe C : Liste des tableaux **102**

Annexe D : Contenu de DVD **103**

Annexe E : Ressources utiles **104**

Annexe F : Exemple des questions **105**

Annexe G : Outils de TALN **107**

Chapitre 1

Introduction

Au fil du temps, les études scientifiques ont considéré le langage naturel (la langue humaine) comme une particularité humaine. Mais à quoi sert la langue humaine? La langue n'est pas seulement un simple moyen de communication pour transmettre des idées et des sentiments, elle permet en même temps de décrire l'histoire d'un pays ou le système de valeurs d'une nation. « *Elle permet ainsi de conserver et de déterminer son identité et son existence, tel que la vie d'une nation est intimement liée à la vie de sa langue* » [CHERAGUI, 2010]

Du point de vue scientifique, le langage naturel a été l'objet de recherche pour diverses disciplines : la linguistique (la science de la langue), la logique, la philosophie, la psychologie, l'informatique. Chacune l'aborde sous un point de vue différent. Par exemple, pour la linguistique, la langue est un phénomène en soi, en philosophie, la fonction du langage est d'exprimer la pensée en la manifestant extérieurement, dans la psychologie, le langage est l'une des grandes fonctions humaines pour réaliser la communication, en intelligence artificielle, la langue humaine est un moyen de communication entre homme et ordinateur.

L'article [RAO et al., 2010] présente le but du *traitement automatique du langage naturel (TALN)*: réaliser la communication entre les hommes et ordinateurs sans mémoriser des commandes et des procédures complexes. Le TALN est la technique qui permet à l'ordinateur de comprendre le langage naturel utilisé par l'homme. Le langage naturel est un système dont l'apprentissage et l'utilisation sont aisées pour les personnes, mais il s'en avéré plus difficile à maîtriser pour un ordinateur. Aujourd'hui les ordinateurs sont très performants, mais « n'ont pas les compétences langagières d'un enfant de 5 ans. Les langues naturelles sont des systèmes vivants qui changent, interagissent, se transforment. »[TELLIER, 2010]

Utilisant des intonations, des métaphores, des comparaisons ou synonymes l'homme réalise une transmission spéciale de l'information. Le langage naturel est direct, expressif, concret, mais dans le même temps confus, particulier et intuitif. Pour cette raison le langage naturel ne peut pas être compréhensible jusqu'à présent pour l'ordinateur. Malgré les difficultés, le traitement du langage naturel est largement considéré comme une solution prometteuse et très important dans le domaine de la recherche informatique.

Dans le contexte d'aujourd'hui de l'ouverture générale vers un public peu formé à l'informatique et d'autre part du confort d'utilisation, l'intelligence artificielle introduit l'utilisation de langage naturel. On peut faire remonter l'origine au début des années 1950, quand le langage naturel a commencé faire l'objet de recherche en linguistique informatique (ou informatique linguistique). Celle-ci est une discipline à cheval entre la linguistique et l'informatique qui a pour objectif le traitement automatique du langage naturel. Aujourd'hui le traitement du langage naturel devient l'un d'entre les plus actifs domaines de l'interaction homme-machine.

L'objectif général est de donner à l'ordinateur la capacité de comprendre et de générer du langage naturel de sorte que finalement les gens puissent interagir avec les ordinateurs comme s'il s'agissait d'aborder une autre personne. Les applications qui seront possibles sont impressionnantes : les ordinateurs seraient en mesure de traiter le langage naturel, de réaliser les traductions des langues avec précision et en temps réel, ou d'extraire et résumer les informations provenant de diverses sources de données, en fonction des demandes des utilisateurs (traduction et interprétation du [RAO et al., 2010]).

Dans ce mémoire, nous abordons l'interrogation de bases de données en langue naturelle, une application directe du traitement automatique des langues naturelles connu avec le nom de *système de question - réponse*.

Guide de lecture

La suite du chapitre introductif présente les thèmes de notre recherche en quatre sections. La première précise le sujet de notre travail : les systèmes de question-réponse, type particulier de TALN. La deuxième section présente le contexte de notre travail : le logiciel EXPESURF. Notre ambition dans le cadre de ce projet est décrite en section 1.3. Enfin, la section 1.4 situe notre travail par rapport à d'autres travaux. Le chapitre se termine en suite par la présentation de la structure du mémoire.

1.1. Délimitation du sujet

Cette section se base sur le document « Une petite introduction au traitement automatique du langage naturel, support de cours » [YVON, 2007], écrit par François YVON. L'article définit les principaux concepts et les problèmes posés par le traitement automatique du langage naturel. En extrayant de cet article les différentes applications du TALN, nous introduisons le concept de l'interrogation en langue naturelle de base des données, le sujet de notre mémoire.

1.1.1.Traitement Automatique du Langage Naturel

« *On regroupe sous le vocable de traitement automatique du langage naturel l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication.* » [YVON, 2007]

L'origine du traitement automatique du langage naturel se situe dans les années cinquante dans un contexte scientifique de la mise au point du premier traducteur automatique, mais aussi dans un contexte politique d'interprétation, de décodage ou de traduction des phrases russes en anglais, pendant de la guerre froide. À l'époque, le coût de la traduction automatique était deux fois plus cher que la traduction humaine et il donnait des résultats moins bons.

Aujourd'hui, le traitement du langage naturel pose encore des difficultés majeures. Pourquoi est-il très difficile de programmer le langage ? Parce qu'il utilise des concepts et des techniques qui appartiennent à plusieurs disciplines: l'intelligence

artificielle, l'informatique théorique, la logique, la linguistique, mais aussi les neurosciences, les statistiques, etc.

Il y a deux sortes des difficultés rencontrées: de l'*ambiguïté* du langage et de la quantité d'*implicite* contenue dans les communications naturelles. Concernant l'ambiguïté, une caractéristique du langage est cela d'avoir une multitude d'interprétations possibles pour chaque entité linguistique. La communication entre les hommes implique l'existence de la gesticulation, des intonations, des connaissances du monde et de son fonctionnement. Ces éléments du contexte déterminent la disparition de l'ambiguïté et la compréhension de l'énoncé naturel est implicite. Donc une solution consiste à restreindre le contexte d'interactions à un sous-domaine particulier, pour ignorer les ambiguïtés et pour représenter formellement un grand nombre des connaissances nécessaires à la compréhension des énoncés du domaine considéré.

1.1.2. Pensée humaine versus intelligence artificielle

Un ordinateur actuel réalise 10^{17} opérations par seconde. Mais quel est l'avantage de l'homme ? Le plus important est l'élément-surprise : l'homme est imprévisible, sa pensée ne respecte pas toujours un algorithme comme l'ordinateur le fait.

Pour comprendre comment les hommes utilisent le langage, il faut identifier les niveaux de traitement et leurs interactions .

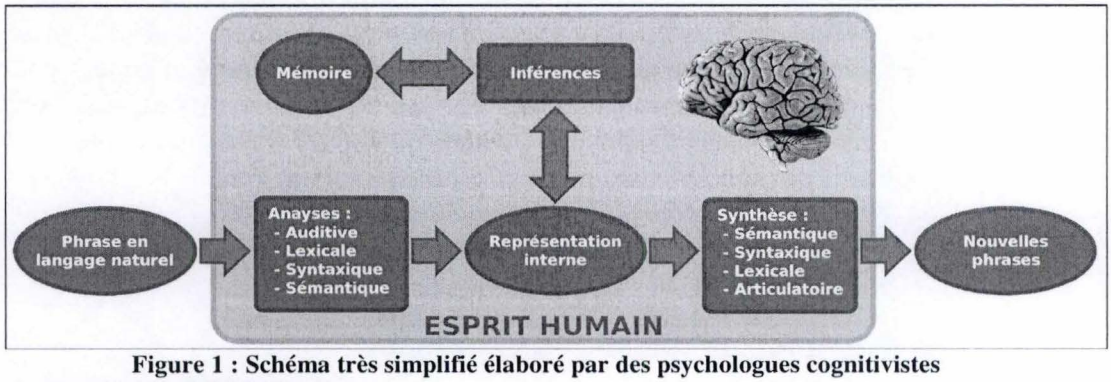


Figure 1 : Schéma très simplifié élaboré par des psychologues cognitivistes
(source [AUDIBERT, 2010])

La Figure 1 représente la façon des psychologues cognitivistes de modéliser le fonctionnement de l'esprit humain. Suivant le schéma, la phrase en langage naturel est transformée en une représentation interne, par une analyse auditive, lexicale, syntaxique et sémantique. C'est cette représentation interne qui peut être mémorisée et manipulée par le raisonnement. Pour générer de nouvelles phrases, il faut transformer cette représentation par l'intermède d'une synthèse sémantique, syntaxique, lexicale et articulatoire.

De l'autre part, si on regarde la « pensée » de l'ordinateur, on observe que pour produire un système qui peut interagir avec l'homme par un langage naturel, les systèmes TALN reproduisent cette architecture, par la traduction des fonctions en

programme. Pour réaliser la compréhension complète d'un énonce, l'ordinateur le soumet à une chaine des traitements :

- traitement phonétique : utilisé seulement dans le cas ou l'entrée du système est un langage vocal pour transformer la voix humaine en une phrase grammaticale ;
- traitement morphologique : reconnait les composantes lexicales et identifie leurs propriétés ;
- traitement syntaxique : identifie des constituants de plus haut niveau et les relations entre eux ;
- traitement sémantique : comprend la phrase en associant à chaque concept évoqué un objet ou une action dans un monde de référence (réel ou imaginaire) ;
- traitement pragmatique : étudie le sens du contexte, trouvant la signification réelle de la phrase liée aux conditions situationnelles et contextuelles.

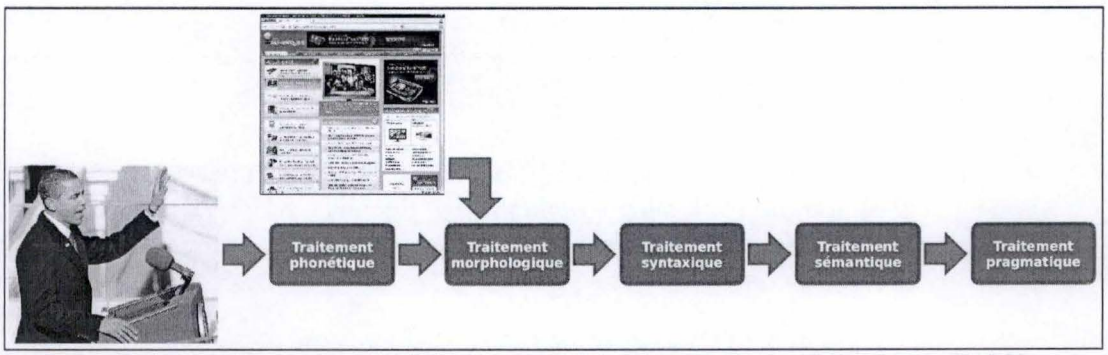


Figure 2 : Architecture séquentielle de la suite des traitements d'une application du TALN

Le Figure 2 présente l'enchainement des étapes qui sont faites par l'ordinateur pour réaliser le traitement du langage naturel vocal. La première étape réalise un traitement phonétique. Elle est suivie par un traitement morphologique pour déterminer les informations grammaticales associées à chaque mot considéré isolé. Ensuite, le traitement syntaxique consiste à extraire les relations grammaticales que les mots et groupes de mots entretiennent entre eux. Le traitement sémantique analyse le sens de la phrase et enfin le traitement pragmatique interprète la phrase en fonction des connaissances générales sur le monde et de la situation de communication.

1.1.3. Les systèmes de questions - réponse

Quelles sont les applications utilisées aujourd'hui, mettant en œuvre les recherches en Traitement Automatique du Langage Naturel ?

L'intérêt pour l'informatique linguistique a cru radicalement grâce à l'évolution de technologies du traitement du langage naturel, du traitement du son, l'interprétation automatique du texte, intelligence artificielle, etc. Le développement de ces technologies a déterminé l'apparition des moteurs de recherches sur Internet, des applications de reconnaissance de la voix, de la traduction automatique, des applications pour le classement automatique du texte. Aussi aujourd'hui les correcteurs pour les éditeurs du texte sont parmi les applications les plus connues utilisées avec succès. Grâce à cette technologie sont apparus les robots pour le téléphone qui portent des conversations, les classeurs automatisés de documents, les systèmes d'extraction des concepts prédominants dans le texte (par exemple identification automatique des personnes, des pays, des villes) ou le Web sémantique.

Aujourd'hui, les applications de TALN sont nombreuses et variées. Elles ne diffèrent pas seulement par leur contexte d'utilisation, mais aussi par le type de données linguistiques d'entrée : textes écrits, dialogues oraux ou écrits et unités linguistiques (mots, phrases, énonces). Selon [YVON, 2010], nous avons regroupé les applications qui traite le langage écrit dans trois grandes catégories : traitement de documents, production de documents et interfaces homme-machines.

Dans ce mémoire nous abordons une application du domaine des interfaces naturelles, l'interrogation en langage naturel de base de données, connue aussi sous le nom de *système de question-réponse*. Les systèmes de question – réponse (QR ou en anglais Question answering systems) sont considérés comme l'étape suivante dans l'évolution du moteur de recherche de l'information. Les systèmes QR font partie du *système d'acquisition des données.* Les systèmes d'acquisition des données sont les systèmes qui obtiennent des données d'une source extérieure.

Les systèmes de question – réponse sont caractérisés par le fait qu'ils reçoivent des questions en langage naturel et par une suite des traitements (phonétique, morphologique, syntaxique, sémantique) les traduisent vers une requête codée dans un langage informatique d'interrogation de base de données, comme SQL par exemple.

1.2. Le logiciel EXPESURF

La recherche du mémoire est orientée vers le développement d'un système d'interrogation en langage naturel au sein du logiciel EXPESURF. Commençons par quelques définitions pour comprendre mieux la notion de *système expert.*

Un système expert est une application qui prend des décisions ou qui résoudre des problèmes d'un certain domaine ayant à la base des connaissances et des règles analytiques établies par les experts. La principale différence entre un système expert et un programme conventionnel est qu'un système expert utilise des connaissances tandis qu'un programme conventionnel exploite des données. Il peut être exprimé succinct par l'équation suivante :

système expert = base de connaissances + mécanisme d'inférence + interface

Equation 1 : La définition d'un système expert

*La base de connaissance*s est l'ensemble des éléments et des états qui constitue la description de l'univers du domaine pour lequel on applique le système expert. *Le mécanisme d'inférence* est une ensemble des procédures avec le but de manipuler la base de connaissances pour effectuer des rationnements sur base du contenu. D'habitude, le système expert utilise une interface en langage naturel pour réaliser une communication conviviale entre utilisateur et système.

Cette section, inspirée du rapport scientifique du logiciel EXPESURF [Rapport 3 EXPESURF], présente le contexte du projet EXPESURF et le besoin d'implémentation d'un système de question-réponse.

EXPESURF est un système expert qui a comme objectif *« de créer un service spécialisé dans la diffusion des informations concernant les traitements des surfaces des matériaux et le choix de ces traitements pour une application industrielle donnée.*

En interrogeant le logiciel expert, les chercheurs, concepteurs et développeurs comparent différentes solutions dont certaines sont innovantes. EXPESURF donne des informations pour définir le traitement à appliquer sur une nouvelle pièce ou pour améliorer des propriétés apportées par le traitement de surface qu'il utilise déjà. Le système propose des solutions pour répondre aux problèmes écologiques soulevés par certains traitements. » ([Rapport 3 EXPESURF])

Le logiciel EXPESURF contient deux processus : AC (Acquisition de Connaissances) et AD (Acquisition de Données). La partie pratique de ce mémoire aborde la partie d'acquisition de connaissance du EXPESURF. Le but du processeur d'acquisition de connaissances (AC) est d'introduire des connaissances dans le système en accédant une base de données. Le système utilise une interface qui permet aux utilisateurs d'extraire des données.

Pourquoi le logiciel EXPESURF a-t-il besoin d'un système de consultation de base de données en langage naturel ?

Dans le contexte du traitement de surface, les exigences fonctionnelles augmentent et les contraintes de l'environnement sont difficiles à respecter durant l'élaboration, en cours d'utilisation et de vie des pièces. En général, un seul traitement n'est pas suffisant, on réalise donc plusieurs couches successives par multitraitements. Comment savoir toutes les interactions et les propriétés de la couche antérieure sans être un spécialiste dans les bases de données du logiciel? L'interface en langage naturel résout ce problème permettant l'ouverture générale vers les clients qui n'ont pas une formation spécifique et le traitement d'une grande quantité d'information dans un délai plus court.

1.3. L'objectif de l'étude

Le titre de ce mémoire « Consultation de base de données en langage naturel, pour le logiciel EXPESURF» souligne l'application du traitement automatique en langage naturel, une discipline en grande évolution aujourd'hui. Les interfaces en langage naturel apparaissent comme résultat de la nécessité d'interfaces de plus en plus ergonomiques et du besoin du traitement (recherche, traduire, analyser, produire) des informations textuelles en format électronique.

Le travail que nous avons élaboré en vue de la présentation de notre mémoire de fin d'études en sciences informatiques consiste à développer et à mettre en œuvre un système de question-réponse pour le logiciel EXPESURF. Il s'agit en effet de l'extraction d'une information précise définie par une question. Une fois le contexte et le type d'information compris en analysant la question, il doit être possible de construire une requête pour la recherche de la réponse dans la base de données.

L'objectif du mémoire est d'une part de réaliser une étude détaillée des systèmes de question-réponse et de la variété des outils existants et d'autre part d'étudier l'interaction des mots dans les propositions interrogatives pour la langue française et de trouver un algorithme pour traduire les questions en langage naturel dans un langage informatique d'interrogation d'une base de données d'un domaine précis.

1.4. Relations avec autres travaux

Notre application est un « plug-in » pour le logiciel EXPESURF, un système expert dans le domaine de surface engineering qui fournie des informations concernant le choix des traitements des surfaces des matériaux à appliquer sur une nouvelle pièce.

Notre solution trouve son origine dans l'article « Une bibliothèque d'opérateurs linguistiques pour la consultation de base de données en langue naturelle » écrit par Béatrice Bouchou et Denis Maurel et présenté au Conférence TALN organisée par ATALA en 1999 ([BOUCHOU, 1999]). Nous avons gardé la même idée de traduire la question en langage naturel vers une requête SQL par l'intermédiaire d'un dictionnaire de mots clefs et d'une bibliothèque des opérateurs. Mais à la différence de l'article qui applique la théorie de Z.S. Harris, pour le choix des opérateurs nous avons utilisé une analyse approfondie de la structure de base de données et de la configuration syntaxique des mots dans une phrase interrogative.

Une autre application très intéressante dans le domaine des systèmes de question-réponse est le projet Chat-80[1]. Chat-80 un système en langage naturel qui permet à l'utilisateur d'interroger une base de connaissances Prolog dans le domaine de la géographie du monde. Fernando Pereira dans sa thèse de doctorat «Logic for natural language analysis » ([PEREIRA, 1983]) utilise la logique formelle comme un outil de décrire la syntaxique et la sémantique pour une partie de la langue anglaise, théorie qui se trouve à la base du logiciel Chat-80. Il est parmi la première grande démonstration d'utilisation du Prolog en traitement du langage naturel.

1.5. Structure du mémoire

Pour atteindre les objectifs fixés pour ce mémoire, nous abordons le sujet en cinq chapitres.

Dans le premier chapitre, l'introduction, nous avons commencé par la présentation du rôle du langage naturel dans la vie des hommes et dans la recherche des plusieurs disciplines. Ensuite, nous avons défini les principaux concepts et les problèmes posés par le traitement automatique du langage naturel et comparé la pensée humaine avec l'intelligence artificielle. Une application directe de ce domaine : les systèmes de question-réponse et notre logiciel de base EXPESURF ont été présentés ainsi que les objectifs et les relations avec les autres articles consultés.

Le chapitre 2 présente une approche plus précise des systèmes de question-réponse. Tout d'abord il présente l'intersection des systèmes QR avec d'autres domaines. Nous approfondissons le domaine d'intelligence artificielle et le domaine TALN en insistant sur les niveaux d'analyse auxquels nous pouvons soumettre le langage et sur les applications existantes aujourd'hui. Ensuite nous parcourons l'histoire de systèmes de QR et nous présentons quelques domaines qui utilisent le langage naturel.

Ensuite, dans le chapitre 3, nous introduisons l'architecture générale du développement des systèmes de question-réponse et les campagnes d'évaluation pour

[1] Chat-80 : http://www.lpa.co.uk/pws_dem5.htm

les systèmes QR. Nous finissons par un passage en revue des outils existants pour le développement d'un système de consultation en langage naturel d'une base de données et nous argumentons notre choix d'implémentation pour le logiciel EXPESURF.

Le chapitre 4 est vu comme le noyau de notre recherche, étant dédié à l'implémentation des systèmes de question-réponse. Au début nous introduisons le système EXPESURF en présentant son objectif, sa structure, la base de données et les entités du schéma de la base de données. Puis nous réalisons une étude de la grammaire de la langue française, en mettant accent sur la structure de la phrase interrogative et en analysant les configurations syntaxiques dans une question en fonction de leur type. Pour l'implémentation d'un système de consultation en langage naturel pour notre logiciel, nous proposons une identification des mots clefs de la question en utilisant une analyse de base de la phrase interrogative pour la langue française.

Le cinquième chapitre est consacré à la conclusion visant à mentionner l'importance du travail effectué et aussi les perspectives futures de notre étude.

Chapitre 2

Système de question – réponse

2.1. Introduction

L'intelligence et l'information ne peuvent pas être séparées l'une de l'autre. Les hommes sont capables de fournir une information utile, possédant l'intelligence, mais ils sont limités du point de vue de connaissances. Les systèmes d'informations qui utilisent des bases de données ont cette compétence, mais ils n'ont pas le raisonnement natif des hommes.

Le volume des connaissances croit dans un rythme qui dépasse notre capacité d'accumulation des informations. Chaque jour apparaissent plus de documents qu'un homme ne peut en lire dans une année de lecture continue. Apprendre quoi, comment et où chercher des connaissances nécessaires pour la résolution d'un problème devient aujourd'hui une étape essentielle d'un processus éducationnel moderne : « une question bien formulée peut contenir un morceau de la solution cherchée ».

Comme précisé plus haut, aujourd'hui il est de plus en plus important de savoir quoi et où chercher. À l'heure actuelle, l'*Internet* contient la plus grande base de connaissances et se trouve dans une continue expansion et renouvellement. L'Internet est un système informationnel qui par l'interaction avec les agents humains construit une base de connaissances en langage naturel. En même temps, il est le lien le plus accessible où les informations peuvent être consultées. Par contre, un désavantage de l'Internet est qu'à cause de la multitude des informations disponibles il est difficile de trouver des informations nécessaires.

Les méthodes les plus efficaces de découverte et d'acquisition des informations sont les *moteurs de recherche*. Le but des moteurs de recherche est d'offrir à l'utilisateur un ensemble des articles et des pages Web avec l'information nécessaire. Un possible désavantage est cela qu'ils n'offrent pas une réponse satisfaisante ou concrète, mais seulement un ensemble de sites, d'où l'utilisateur doit extraire tout seul l'information cherchée.

Un pas suivant dans ce domaine est le développement des systèmes capables de répondre aux informations décrites par l'utilisateur en langage naturel. Le but est d'assurer une réponse à la question d'utilisateur qui accomplir les trois conditions : être correct, formulé en langage naturel et concis. Un système de question-réponse nécessite un traitement du langage naturel complexe.

Le traitement du langage naturel est un sujet très attractif grâce à son applicabilité dans les domaines comme l'interaction homme-machine. Dans la pratique, on observe des difficultés majeures à cause de diversité des interprétations des affirmations en langage naturel et de multitudes des sens dont les mots peuvent l'avoir. Les systèmes des questions-réponses qui sont un sous-domaine de TALN, héritent de ce problème.

La problématique des systèmes de question – réponse (SQR) se situe à l'intersection de plusieurs domaines, comme la recherche d'informations, le traitement automatique de la langue naturelle, l'interaction homme-machine, l'intelligence artificielle, présentée dans la Figure 3.

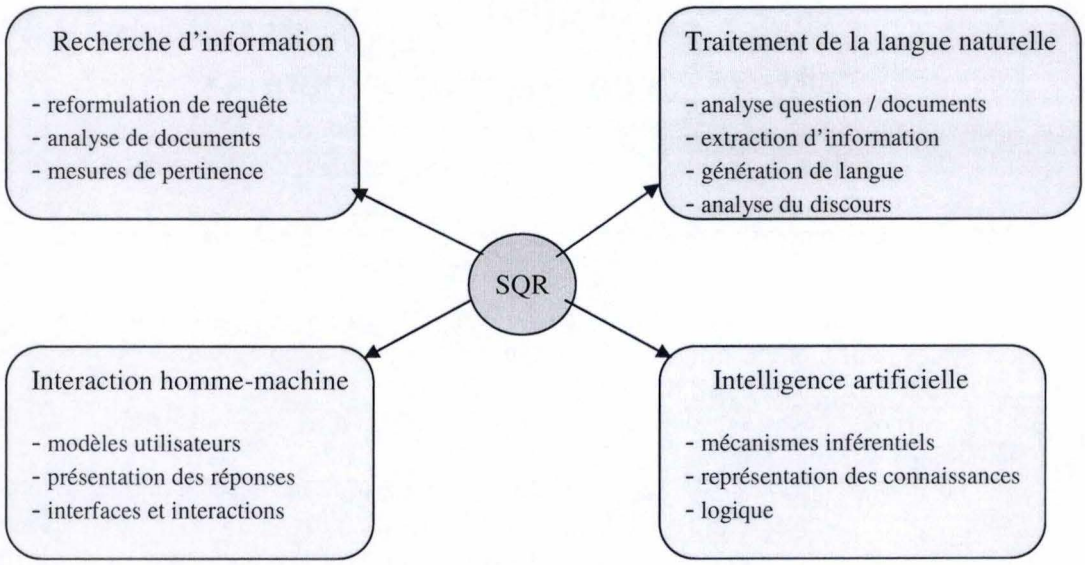


Figure 3 : Les interactions des systèmes des questions-réponses avec autres domaines (source [MORICEAU, 2009], page 6])

Ce chapitre présente une nouvelle approche de systèmes de question-réponse. Au début, nous introduisons le domaine d'intelligence artificielle en présentant les principaux techniques et quelques applications qui les utilisent. Ensuite nous mettons accent sur le traitement automatique du langage naturel. Nous analysons les niveaux de traitement auxquels nous pouvons soumettre le langage naturel et les applications existantes aujourd'hui. Ensuite nous parcourons l'histoire de systèmes de QR et nous présentons quelques domaines qui utilisent le langage naturel. Nous conclurons ce chapitre par un passage en revue des articles utilisés et de notre contribution.

2.2. Intelligence artificielle

Dans cette section, en consultant les articles [GIONEA, 2008] et [VASILESCU, 1996] nous introduisons l'évolution de l'intelligence artificielle au fil du temps et les principales techniques et applications utilisées aujourd'hui.

Est-ce qu'on peut remonter l'origine de l'intelligence artificielle au début de la création des premiers ordinateurs ? Ont-ils la capacité de pensée ?

À partir de cette question l'article [GIONEA, 2008] parle de l'intelligence des machines comme le résultat de plusieurs années de recherche, tests, réussit et échecs. Au cours des années, on a voulu que les machines apprennent, comprennent notre langage transmis par l'intermède des interfaces et perfectionnent la perception sensorielle.

Au début de l'Intelligence Artificielle en 1956, tout semblait une utopie, un rêve trop beau pour être atteint. Au cours des presque 50 dernières années, le terme a pris contour, devenant réalité. Maintenant, il est utilisé dans toutes les sciences qui veulent s'affirmer. Son initiateur, le professeur John McCarthy, a présenté le nouveau concept en 1956 lors de la réunion "Projet de recherche d'été sur Darthmouth Intelligence Artificielle".

L'intelligence artificielle peut être définie comme la simulation d'intelligence humaine traitée par des machines, des systèmes informatiques en particulier. Le terme d'intelligence artificielle est rencontre aujourd'hui dans nombreux publications scientifiques quand il s'agit des applications comme reconnaissance et l'analyse de la voix et des images, des traductions des langues ou des divers jeux d'intelligences (comme bridge). Initialement les objectifs de l'intelligence artificielle ont été très ambitieux : la machine devait résoudre divers problèmes, apprendre de son expérience et d'événements extérieurs au système, effectuer des raisonnements, concevoir de nouveaux objectifs avec des propriétés prédéfinies.

Le principal but de l'Intelligence Artificielle est d'imiter entièrement le cerveau humain, dans la manière dans laquelle il pense, répond et interagit. Les évolutions des recherches d'aujourd'hui montrent que ce but ne peut pas être réalisé prochainement, à cause du fait que le cerveau reste encore une énigme, presque impossible à analyser mathématiquement et/ou traduite en langage machine.

Dans les résultats des recherches de dernières années, on observe que l'ordinateur est capable de réaliser les raisonnements et de découvrir les liens logiques entre les événements décrits par les propositions. Aussi l'ordinateur est capable d'apprendre de ses propres erreurs et d'interagir avec un utilisateur. Utilisant ces performances l'homme a créé des ordinateurs et des programmes pour travailler pour lui, pour résoudre des équations compliquées, pour traiter des bases de données avec beaucoup d'enregistrements, pour projeter et produire des équipements techniques avancés. Mais la limite est loin d'être atteinte, les chercheurs étant préoccupés de la réalisation de « la machine qui pense » et qui peut offrir instant des solutions viables aux diversités des problèmes qui apparaissent.

Sous un autre angle, l'article [VASILESCU, 1996] présente l'idée du mathématicien anglais, Alan Turing, qui considère qu'un ordinateur est intelligent s'il peut être confondu avec un être humain. Le test qui porte son nom est parti du « jeu d'imitation » auquel il participe trois acteurs : une machine (A), un homme (B) et un autre homme (C). Il a comme contraintes : A et B ne sont pas dans la même chambre avec C, C ne connaît pas quel d'entre les deux premiers participants est homme et quel est machine et il ne peut pas les entendre ou les voir. La communication se réalise seulement par l'intermède d'un terminal. Le but de C est d'observer la différence entre l'homme et ordinateur en fonction des réponses reçues.

Le test Turing symbolise l'idéale de l'intelligence artificielle comme partie de l'informatique. Pour résister à un test humain, l'ordinateur doit stocker une quantité immense d'informations de tous les domaines. Turing considère que pour accomplir ce but la solution n'est pas la programmation de l'ordinateur, mais plutôt l'éducation d'une machine – enfant capable d'apprendre de son expérience et d'utiliser un langage naturel pour enrichir ses connaissances.

Parmi les principales techniques de l'Intelligence Artificielle, nous mentionnons les suivantes.

∞ Systèmes expert

Un système expert est un programme qui analyse des connaissances et fait des rationnements pour obtenir des résultats aux problèmes difficiles. Les informations reçues de l'ordinateur sont semblables à celles données par un expert humain dans son domaine. Du point de vue fonctionnel, le système expert utilise une base de connaissances et un algorithme de recherche spécifique à la méthode de raisonnement.

Le raisonnement et les connaissances ne doivent pas être traités séparément, parce que ce type d'application réalise l'harmonisation entre elles.

En comparaison avec les autres programmes de calcul qui demandent des informations complètes pour prendre des décisions, les systèmes expert ont été développés pour trouver la solution optimale utilisant la base de données disponible.

La diversité des domaines d'application des systèmes experts est remarquable : en commençant par l'architecture, l'archéologie, la banque, le commerce, l'éducation jusqu'à l'ingénierie des systèmes et la médecine.

Parmi les avantages offerts, notons: la rapidité, consistance et efficience par rapport aux experts humains, synthèse d'informations provenant des plusieurs experts, toujours disponibles et infatigabilité.

∞ Réseaux neuronaux

Les réseaux neuronaux sont des systèmes qui simulent l'intelligence par la reproduction des types de connexions physiques qui se trouvent dans le cerveau humain. À cause des limitations technologiques, le nombre de ces connexions est très petit en comparaison avec les connexions du cerveau humain. Même si le fonctionnement est semblable avec cela humain, la structure des réseaux neuronaux est différente de celle-là du cerveau. Un réseau neuronal est plus simple que le correspondant humain, mais il est aussi composé des unités de calculs puissantes, mais inférieures au neurone.

∞ Agents intelligents

Les agents sont des entités qui fonctionnent autonome, accomplissent certains buts et interagissent avec des opérateurs humains ou avec autres agents. Le fonctionnement autonome suppose un milieu dans lequel ils travaillent. Par exemple dans le contexte du web le milieu inclut : des documents existants sur le Web, des programmes qui s'exécutent sur le web, des utilisateurs humains et des autres agents. Dans ce milieu, l'agent accède aux informations et exécute des programmes en écrivant des documents et communiquant avec les autres. Par exemple, il collectionne des nouvelles de sites Internet, envoyant des messages e-mail ou filtrant les messages reçus.

Même s'ils travaillent avec des mots clés et ils sont encore l'objet des recherches, les agents vont devenir très utiles par exemple pour aider l'utilisateur à trouver seulement les articles qui l'intéresse, lui faisant ainsi gagner du temps.

Ensuite nous présentons quelques applications qui utilisent les techniques décrites ci-dessus.

∞ Interfaces naturelles

Les interfaces naturelles sont applications qui impliquent des recherches dans le domaine linguistique, philosophique, sciences des ordinateurs et autres avec le but d'assurer une communication naturelle dans un langage habituel avec l'ordinateur.

Le terme de communication est utilisé pour décrire toutes les procédures par lesquelles une personne influence le cerveau de l'autre personne (communication écrite, orale ou langage du corps). La théorie d'information met accent sur trois dimensions importantes dans l'évaluation de la communication : dimension technique (l'exactitude), dimension sémantique (la précision), dimension effective (l'effet d'information reçu). Une autre caractéristique de la communication est la redondance,

la répétition d'une partie du message. Celle-ci peut avoir un effet positif croissant l'efficience du système si le message est bien compris, ou un effet négatif si sont transmises plus d'informations qu'il n'est nécessaire.

∞ Robots

L'intelligence artificielle, l'ingénierie et la philosophie sont les disciplines de base de la robotique. Celles-ci permettent la construction des machines qui disposent des systèmes expert et sont contrôlées par des ordinateurs exécutant des activités humaines.

Les nouveaux modèles de robots ont dans leur composition des ordinateurs qui peuvent entendre, voir et réagir aux divers stimule. Il y a déjà des robots qui marchent comme les hommes, qui peuvent faire la différence entre plusieurs voix, s'orientent dans l'espace, reconnaissent des objets, choisissent le chemin plus court entre deux points et évitent les obstacles.

∞ Jeux sur l'ordinateur

Le développement des jeux dans le domaine multimédia est dans grande expansion. À l'heure actuelle, tous les jeux ont dans leur structure des éléments d'Intelligence Artificielle.

Comme le montre la Figure 3 les systèmes de QR utilisent des stratégies issues de l'Intelligence Artificielle :

- mécanismes d'inférence

  Le mécanisme d'inférence est une ensemble des procédures ayant pour but de manipuler la base de connaissances d'un système de règles pour effectuer des raisonnements sur base du contenu. Le système de question-réponse utilise une forme de mécanisme d'inférence : à partir d'un but il le décompose en sous-problème et à chaque pas il sélectionne une règle pour atteindre au moins un des objectifs à résoudre.

- représentation des connaissances

  Conformément à la définition du Alan Newell, l'un des pionniers et théoriciens de l'intelligence artificielle, « *la connaissance est ce qui peut être attribué à un agent humain ou artificiel (un programme de l'ordinateur autonome qui interagit avec son environnement) de sorte que son comportement puisse être catalogué comme rationnel* » [NEWELL, 1982]. Les connaissances utilisées par le système sont groupées dans une base de connaissances et représentent l'expérience accumulée par les spécialistes humains pendant la résolution des problèmes du domaine. Il s'agit d'une ontologie du domaine, des règles utilisées, des restrictions qui restreignent l'espace de recherche.

  Dans l'intelligence artificielle sont utilisées plusieurs représentations structurées des connaissances : réseaux sémantiques, frames, logiques terminologiques, graphes conceptuels.

- logique

  À partir de concepts de connaissance et représentation des connaissances introduites plus haut, on présente les formalismes de représentation : logique des propositions (pas de quantificateurs et pas des

variables), logique de premier ordre (introduction de variables et des quantificateurs) et les règles de production.

Par exemple pour la proposition « Le matériau Fer a la propriété dureté. », on peut la représenter dans la logique des propositions sur la façon suivante :

materiau (FER)
est-propriete (FER, DURETE)

Et pour la proposition « Un matériau a la propriété dureté. » la logique de premier ordre est :

materiau(X)
est-propriete(X, DURETE)

2.3. Traitement automatique du langage naturel (TALN)

Il est bien connu que la plus importante propriété d'une interface d'un système est la qualité d'être conviviale. Qu'est que c'est plus naturel dans une interface homme-machine qu'une communication réalisée dans un langage naturel ?

Le domaine d'intelligence artificielle qui se préoccupe de cet aspect est « le traitement automatique du langage naturel », une discipline à la frontière de la linguistique et des sciences cognitives. En ce qui concerne le langage naturel, le principal but de l'intelligence artificielle est de permettre la réalisation de la communication homme-machine sans être nécessaire à mémoriser des commandes et des procédures complexes.

L'homme de science Noam Chomsky est considéré comme le père du traitement du langage naturel est considéré. À la fin des années 1950, il publie les travaux sur la syntaxe des langages naturels et sur les relations entre grammaires formelles et grammaires naturelles. L'ouvrage « Syntactic Structures » ([CHOMSKY, 1957]) met l'accent sur les structures mentales nécessaires pour représenter le type de connaissances linguistiques indispensables pour parler. En informatique, il est devenu célèbre par « la hiérarchie de Chomsky ». C'est une hiérarchie d'inclusion des classes des grammaires formelles (nommées aussi les grammaires des structures des phrases) qui génèrent les langages formels.

Dans la sous-section 1.1.2 nous avons passé en revue la chaine de traitement du langage naturel. Dans la sous-section 2.3.1 nous traitons en détail ces niveaux de traitements utilisés pour réaliser la compréhension complète d'un énonce. L'article [YVON, 2007] parle de traitement de « bas niveau », traitement lexical, traitement syntaxique, traitement sémantique et traitement pragmatique. En plus, les applications qui utilisent l'oral doivent réaliser un traitement phonétique. Dans la pratique on peut lever des ambiguïtés à un niveau en utilisant les niveaux supérieurs. Dès lors, bien que conceptuellement les niveaux de traitement soient dissociés, ils sont aussi d'une certaine manière couplés.

Ensuite dans la sous-section 2.3.2. nous énumérons quelques applications du TALN qui interagissent avec les systèmes des question-réponses : l'apprentissage automatique, la traduction automatique, la correction orthographique, la synthèse de la parole.

2.3.1. Les étapes du TALN

Traitement phonétique

Selon [HRISTEA, 2000], nous définissons la phonétique la science qui s'occupe de l'étude des phonèmes (entités abstraites qui sont réalisées par une infinité des sons). Les connaissances phonétiques sont cruciales pour le traitement de la parole.

Certaines d'entre les applications de traitement du langage naturel ont en entrée une voix humaine. Pour celles-ci on applique un traitement phonétique. Il s'agit de transformer la voix humaine en une phrase grammaticale.

Selon l'article [L'HAIRE, 2000] il existe deux familles de reconnaissances de la parole et deux modes de reconnaissance : « *Les reconnaisseurs peuvent accepter soit des phrases complètes prononcées de manière normale, soit des phrases avec chaque mot prononcé séparément. De plus, les reconnaisseurs travaillent soit en mode monolocuteur, soit en mode multilocuteur.* » La plus simple variante est celle-là où les mots sont prononcés séparément. Une bonne reconnaissance est déterminée de la qualité de l'entrée sonore et du manque de bruit de l'environnement.

Pour résoudre l'ambigüité, causé par les homophones[2] ce traitement et suit d'une analyse syntaxique.

Traitement de « bas niveau »

Le premier pas du traitement d'une phrase en langage naturel écrit est la segmentation. C'est une étape essentielle qui traduit le texte dans une séquence des unités lexicales (mots).

On distingue plusieurs types des séparateurs identifiés au-dessous:

Séparateur	Signification
.	indique la fin d'une proposition ou une abréviation
,	indique la séparation de deux propositions ou de la partie décimale et numérique
-	sépare des mots composés
'	marque la présence d'une élision, mais peut être utilisé aussi dans les notations du temps (15 minutes : 15').
« »	signale le début ou la fin d'une citation
( )	indique le début ou la fin d'un commentaire
l'espace	sépare les mots consécutifs dans une proposition

Figure 4 : Type de séparateurs

Notre application, décrite dans le chapitre 4, utilise comme séparateur l'espace.

[2] Homographes sont les mots qui s'écrivent différemment et se prononcent de la même manière

Traitement lexical

La morphologie est la science dédiée à l'étude du mot du point de vue de la variation de la forme pour exprimer les diverses catégories grammaticales.([HRISTEA, 2000])

Selon l'article [YVON, 2007] le traitement lexical a comme objectif d'identifier les composants lexicaux et leurs propriétés syntaxiques. Le composant qui effectue cette analyse lexicale s'appelle analyseur lexical ou scanner. Ceci considère le texte d'entrée formé des séquences des chaines des caractères, les reconnaît et les traduit dans les atomes lexicaux appelés *tokens*, en ignorant les éléments non pertinents du texte comme les espaces blancs. Un token est une séquence des caractères qui représente une seule entité dans la grammaire du langage.

Les informations morpho-lexicales sont identifiées en utilisant des bases de connaissances comme : un lexique ou un dictionnaire.

Si on prend l'exemple « Quelle est la propriété du matériau Cuivre ? », l'étape du traitement lexical va retourner comme résultat :

Mots clefs	Traitement lexical
quelle	pronom interrogatif, féminin singulier
est	verbe, 3pers. singulier, indicatif présent
la	déterminant, féminin singulier
propriété	nom féminin singulier
du	déterminant, masculin singulier
matériau	nom, masculin singulier
Cuivre	nom propre
?	ponctuation interrogative

Figure 5 : Exemple de traitement lexical

Traitement syntaxique

Nous présentons le traitement syntaxique en consultant le document [HRISTEA, 2000].

La syntaxe étudie la combinaison des mots (proposition, phrase) et les fonctions accomplies par ces-ci. Les connaissances syntaxiques traitent la façon d'arrangement des mots pour former des propositions correctes et déterminent quel est le rôle de chaque mot dans la proposition. En 1957, Noam Chomsky introduit la notion de *grammaire générative*, qui décrit les propositions fournissant des règles de constructions. Ces règles sont devenues un standard en linguistique, mais aussi en informatique dans la réalisation des compilateurs. Cette théorie dit qu'un ensemble fini des règles peut décrire un nombre infini de propositions.

Le processus de reconnaissance d'une structure d'une proposition par un ordinateur s'appelle *parsing*. Le traitement du niveau syntaxique est dédié à l'analyse

de la structure des phrases. L'analyseur syntaxique, appelé *parser*, reçois une chaine des tokens et les regroupent dans les constructions correctes du point de vue scientifique.

L'article [YVON, 2007] affirme que le résultat de l'analyse syntaxique est représenté sous la forme d'un arbre, qui met en évidence les dépendances entre les constituants. D'habitude les éléments dans la structure arborescente les éléments sont utilisés sous forme des abréviations (Figure 6).

Eléments d'un arbre de dépendance	**Signification**
DP	le domaine du déterminant (*Determiner Phrase*)
NP	le domaine du nom (*Noun Phrase*)
AP	le domaine de l'adjectif (*Adjective Phrase*)
PP	le domaine de la préposition (*Prepositional Phrase*)
VP	le domaine du verbe à l'infinitif ou au participe (*Verb Phrase*)

Figure 6 : Les éléments d'un arbre de dépendance

Exemple tiré du domaine EXPESURF :

Pour la question « Quelle est la propriété du matériau Cuivre ? » l'arbre de dépendance est présenté dans la Figure 7.

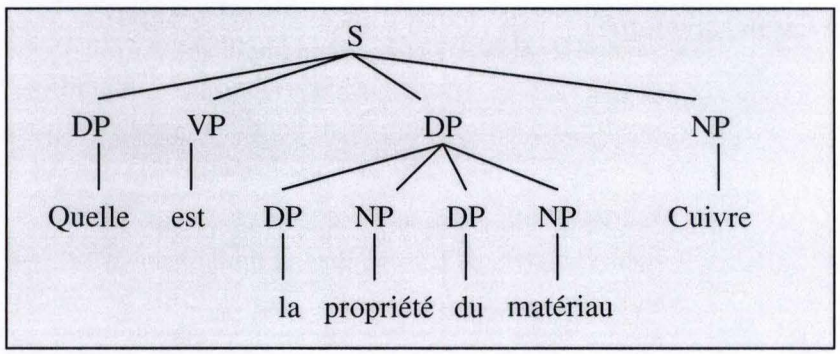


Figure 7 : L'arbre de dépendance pour la question "Quelle est la propriété du matériau Cuivre?"

Le langage naturel contient des phrases ambigües, pour lesquelles sont nécessaires plusieurs analyses. Par exemple on a des mots qui peuvent être soit substantif, soit adjectif, soit verbe. Les hommes ont la capacité des résoudre ces problèmes par l'association avec le contexte. L'ordinateur n'est pas capable sans l'ajout d'une étape de traitement sémantique et pragmatique.

Traitement sémantique

La sémantique est la branche de la linguistique qui l'étude les significations en examinant le sens des mots, des expressions et des phrases, ainsi que leurs relations. Le niveau de traitement sémantique utilisé par TALN s'intéresse au sens des phrases considérées individuellement. Il comprend la phrase en associant à « *chaque concept*

évoqué un objet ou une action dans un monde de référence (réel ou imaginaire). » ([ROZENKNOP, 2010])

L'analyse sémantique peut être réalisée dans le même temps avec l'analyse syntaxique, en validant les expressions correctes du point de vue syntaxique tenant compte des leurs significations et des relations existantes entre les mots.

Pour réaliser cette analyse sémantique, les systèmes de traitement du langage naturel ont besoin de posséder une base de connaissances nécessaire aux traitements. L'article [YVON, 2007] affirme que les informations sémantiques sont identifiées à l'aide des *réseaux*. Le réseau sémantique est une notation graphique pour représenter les connaissances taxonomiques concernant des objets ainsi que leurs propriétés. Les nœuds représentent les concepts et les arcs les relations entre eux.

Une autre base de connaissance utilisée dans l'analyse sémantique est l'*ontologie*. Selon [TRAUSAN, 2004], une ontologie est le résultat d'une expérience vécue qui peut se répéter avec régularité. Elle contient des catégories et concepts fondamentaux, leurs propriétés, leurs relations et leurs distinctions. Les ontologies lexicales incluent un nombre très grand des concepts, liés par un nombre réduit des liaisons. La plus célèbre ontologie très utilisée aujourd'hui est WordNet[3]. Celle-ci peut être vue comme un immense réseau sémantique où les concepts sont représentés par des ensembles des synonymes.

À l'heure actuelle, il existe peu de vrais moteurs sémantiques, car ce type d'analyse consomme trop de ressources. Grâce à l'analyse sémantique, les systèmes des questions-réponses comprennent mieux nos questions, savent évaluer nos besoins et peuvent communiquer plus clairement avec nous.

Traitement pragmatique

La pragmatique examine les effets du contexte sur la production et la réception des énoncés. Elle traite l'utilisation des propositions dans diverses situations et aussi la façon dans lequel le contexte influence l'interprétation d'une proposition. ([HRISTEA, 2000])

Selon l'article [YVON, 2007], le traitement pragmatique étudie « *les attitudes (vérité, désirabilité, probabilité) que les locuteurs adoptent vis-à-vis des énoncés et les opérations logiques que ces attitudes déclenchent* ». Pour la bonne compréhension d'un texte, il est besoin de connaître des éléments qui ne sont pas exprimés explicitement dans le texte : connaissances relatives à la culture générale, au sujet abordé, etc.

La détermination du bon contexte a un rôle important dans les applications comme la traduction automatique, l'interface homme-machine, les systèmes d'aide contextuelle ou d'enseignement.

[3] WordNet est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton avec le but de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise

2.3.2. Applications de TALN

Jusqu'ici nous avons parlé des niveaux de traitement implémentés par les applications TALN pour réaliser une bonne compréhension du langage naturel. Maintenant nous énumérons quelques applications du TALN : la correction orthographique, la traduction automatique, l'apprentissage automatique, la communication homme-machine en langage naturel, le web sémantique, la synthèse de la parole, les systèmes de recherche d'informations, etc. Selon l'article [YVON, 2010] nous avons regroupé les applications qui traitent le langage écrit dans trois grandes catégories : traitement de documents, production de documents et interfaces homme-machine. Les systèmes de question-réponse sont une application directe du traitement du langage naturel, faisant partie de la troisième catégorie.

Passons maintenant aux principales applications du TALN qui interagissent avec les systèmes des question-réponses.

Apprentissage automatique

Helbert Simon a dit que *« l'apprentissage est le processus par lequel un système améliore ses performances »*. Donc on pourrait énoncer que l'apprentissage automatique a comme but la création de programmes qui améliorent les performances des résolutions des problèmes, ayant à la base des données et des résolutions antérieures réalisées.

Selon les articles [HARMACH et al., 2003],[HUANG, 2006] et [PISTOL, 2011], un système d'apprentissage automatique utilise plusieurs méthodes et techniques d'apprentissage :

- L'apprentissage supervisé
  L'apprentissage supervisé est un type d'apprentissage automatique où on évalue les nouvelles instances du problème par une fonction qui produit automatiquement des règles à partir d'un ensemble des exemples.

- L'apprentissage non-supervisé
  L'apprentissage non-supervisé est une méthode d'apprentissage, appelée aussi *clustering*, dans laquelle aucun expert n'est requis. L'objectif est de permettre une extraction de connaissance organisée à partir de ces données d'entrée.

- L'apprentissage semi-supervisé
  L'apprentissage semi-supervisé est une technique d'apprentissage automatique situe entre l'apprentissage supervisé et l'apprentissage non-supervisé. Cette méthode améliore la qualité de l'apprentissage par l'utilisation d'un ensemble de données étiquetées et non-étiquetées dans le même temps.

Aujourd'hui, l'apprentissage automatique est utilisé pour doter les systèmes avec la perception de l'environnement. Parmi les algorithmes utilisés, on énonce les arbres de décision, les réseaux de neurones, les méthodes statistiques, etc. On passe en revues quelques applications : les moteurs de recherche, la robotique, la

reconnaissance de la parole, les sites Web adaptatifs, la détection de fraudes, l'analyse financière, l'aide aux diagnostics.

Traduction automatique

Une autre application importante est la traduction automatique. Celle-ci permet aux hommes de communiquer librement indépendamment de leur nationalité et la langue parlée.

On peut remonter l'origine de la traduction automatique pendant à la guerre froide. Les systèmes de traduction automatique traduisent les interrogations ou toute la base de données dans une autre langue. Il y a deux catégories d'outils de traduction automatique : les outils gratuits en ligne et les logiciels professionnels. Le plus connu outils de traduction gratuite disponible en ligne est celui réalisé par Google : Google Translate[4]. De l'autre côté, les logiciels professionnels ne traduisent pas seulement des pages Web, mais aussi des présentations et des fichiers PDF. Par exemple SYSTRAN développé depuis 40 ans des traducteurs automatiques, associe aux dictionnaires linguistiques intelligents, qui lui permettent de garantir la qualité de ses traductions.

Le Journal de la Traduction[5] précise dans l'article « Traduction humaine et traduction automatique » ([DAUBEY, 2010]) que la traduction réalisée par les hommes et celle réalisée par les ordinateurs peuvent être comparées : « *il s'agit de deux processus complètement différents, tant en termes de processus, de coût que de résultats* ». Les outils de traductions les plus performantes ne peuvent pas maitriser une langue aussi bien qu'une personne de langue maternelle. L'article propose un petit test : « *Saisir un texte en français, le traduire en anglais via un logiciel de traduction et procéder ensuite à une traduction en français, cette fois, du texte anglais. Le résultat est sans appel, le texte traduit n'a rien en commun avec le texte original, il est de plus incompréhensible.* »

Correction grammaticale

Les correcteurs grammaticaux sont une application du TALN très utilisé aujourd'hui. Ils ont comme but l'amélioration de la production du texte. L'article [YVON, 2007] identifie plusieurs types des correcteurs :

- les claviers « auto-correcteurs » ;
- les correcteurs d'orthographe ou de syntaxe ;
- les correcteurs stylistiques ou les aides intelligents à la rédaction .

Les correcteurs proposent une série des mots pour corriger les erreurs d'accentuation ou l'écriture phonétique du mot, en vérifiant en même temps la structure de la phrase.

Synthèse de la parole

Les systèmes de la synthèse de la parole ont comme but de transformer un texte en sons. Le son reste un peu métallique et un peu artificiel, mais grâce à la qualité des ordinateurs actuels il est intelligible pour l'oreille humaine.

---

[4] Google translate site: http://translate.google.be/

[5] Journal de la Traduction : http://blog.atenao.com/

L'article [L'HAIRE, 2000] passe en revue les étapes utilisées par l'ordinateur pour prononcer une phrase. Une première étape est l'analyse de la phrase, qui résout aussi les ambigüités déterminées des homographes hétérophones[6]. Cette étape est suive d'une d'analyse syntaxique. En comparaison avec l'être humain, l'ordinateur ne sait pas comment utiliser l'élément de prosodie[7]. Par exemple une phrase interrogative ou exclamative utilise une intonation différente de celle d' une phrase déclarative.

Il existe deux méthodes pour prononcer une phrase. La première utilise une concaténation des diphones, ça veut dire que les phrases, les mots et les syllabes sont formés par la collation des petites unités sonores pour varier l'intonation. La deuxième méthode, nommée synthèse par formats, modélise le conduit vocal (langue, dents, lèvres, glotte, luette, palais) et simule par des paramètres la production de sons. ([L'HAIRE, 2000])

Cette application est utilisée pour apprendre des langues étrangères, pour permettre aux hommes qui ne peuvent pas parler de communiquer aux téléphones ou par des systèmes de guidage automobile

2.4. Système de question-réponse

Les systèmes de question – réponse sont une application directe du traitement automatique du langage naturel, domaine de l'intelligence artificielle qui permet la réalisation de la communication homme-machine en langage naturel en facilitant l'accès de l'information au grand public.

Cette section décrit les systèmes de question-réponse comme de systèmes qui implémentent une interaction homme machine et réalisent une recherche d'information. Au début de la section, nous parcourons l'histoire de systèmes de question-réponse, en donnant comme exemples de logiciels connus : LUNAR, ELIZA, MYCIN et SHRDLU. Les systèmes de question – réponse sont considérés comme l'étape suivante dans l'évolution du moteur de recherche de l'information. La sous-section 2.4.3. présente les similitudes entre les étapes du traitement du langage naturel réalisées par les systèmes de la recherche d'information et celles de systèmes de question-réponse.

2.4.1. Histoire

Les années 1965-1975 représentent la période de l'intelligence artificielle où le monde se préoccupe de la « compréhension », ça veut dire qu'elle veut faire en sorte que les machines comprennent le langage naturel.

Les premiers systèmes de question-réponse ont été des interfaces en langage naturel pour interroger des systèmes expert crées pour les divers domaines. Un système efficient à l'époque est le logiciel LUNAR (Woods, 1977) qui accède des données du domaine chimique correspondantes aux roches trouvées lors des missions Apollo. Dans une conférence en 1971, LUNAR a réussi à répondre à 90 % des

[6] Homographes hétérophones sont les mots qui s'écrivent de la même manière, mais se prononcent différemment.

[7] La prosodie est le système de prononciation des syllabes ou d'intonation de la voix dans une phrase

questions posées ([VASILESCU, 1996]). Les principales caractéristiques de ces systèmes sont que les utilisateurs n'ont pas été formés pour travailler avec le système et que les informations sont stockées dans une base de données réalisée par des experts du domaine respectif.

Un autre programme de question-réponse est SHRDLU. Il comprend les commandes données en anglais pour modifier un ensemble des cubes et il répond aux questions liées à la configuration des pièces. Par exemple, l'ordinateur sait répondre à une question de type « Quelle couleur a la pièce au-dessous de la pyramide rouge ? ». Le programme exécute aussi des commandes de type « Mettez la pyramide bleue au dessous de la pièce vert ». Mais ce monde des pièces est très simple et SHRDLU n'a pas été très utile dans la résolution des problèmes concrets. ([GENTHON, 2004])

L'article [VASILESCU, 1996] parle d'une autre application spécialisée le domaine de la médecine : le programme MYCIN. Ce logiciel propose de diagnostiquer les maladies infectieuses du sang et de recommander un traitement. Pour atteindre le même objectif, les hommes ont besoin de beaucoup des connaissances. Le programme a été plus utile que les autres programmes. Un test qui compare le diagnostic donné du MYCIN avec celui donné par des médecins des diverses spécialités a choisi comme la solution celle présentée par MYCIN. Même s'il n'a pas été utilisé effectivement, le programme a ouvert le chemin vers une série des systèmes expert.

Un autre programme fameux de l'époque est ELIZA. ELIZA simule le comportement d'un psychologue qui tient une conversation en anglais avec les patients. L'article [HARMACH et al., 2003] présenté un exemple de conversation entre ELIZA est une adolescente.

«
? ELIZA : Bonjour. Qu'est-ce qui vous amène?
| : *Je suis triste.*
? ELIZA : Depuis combien de temps êtes-vous triste?
| : *Depuis le départ de mon frère*
? ELIZA : Pouvez-vous me parler de votre frère?
| : J*'aime discuter avec lui.*
? ELIZA : Quelqu'un d'autre dans votre famille aime-t-il cela?
| : *Non.*
? ELIZA : Continuez
| : *Je me sens seule .*
? ELIZA : Ressentez-vous souvent cela?
| : Oui , depuis que mon frère est parti !
? ELIZA : Pouvez-vous me parler de votre frère?
| : *Vous vous moquez de moi?*
? ELIZA : Continuez
| : B*ye*
? ELIZA : Au revoir. J'espère vous avoir aidé.
»

Les connaissances du programme de l'anglais et de la psychologie ont été codifiées sous forme d'un ensemble des règles simples. Le programme contient un ensemble des mots clés et pour chacun il associe une ou plusieurs règles. Par exemple si la proposition du patient contient le nom « mère » ou « père », ELIZA répond avec

« Parlez-moi de votre famille ». Le procédé d'ELIZA s'appelle « il ne comprend, mais correspond » (« pattern matching » en anglais). Le même article présente une structure de l'algorithme utiliser par le logiciel ELIZA : «

- lire une phrase ;
- choisir une paire (stimulus, réponse) ;
- apparier la phrase et le stimulus ;
- écrire la réponse associée ;
- recommencer. »

Dans les sous-sections suivantes, nous présentons les interactions des systèmes de questions-réponses avec les domaines d'interaction homme-machine et recherche d'information.

2.4.2. Interaction homme-machine

L'interaction homme – machine ou interface homme – machine est un domaine interdisciplinaire qui étudie l'interaction entre utilisateurs et ordinateurs. Grâce à cette interaction, le domaine se trouve à la frontière des sciences exactes (ex. : informatique) et les sciences humaines (ex. : psychologie, linguistique, sociologie). Cette liaison apparaît au niveau des interfaces avec l'utilisateur (en anglais : *user interface*) et inclut des aspects ergonomiques, software et hardware.

Les systèmes des questions-réponses sont des systèmes qui impliquent une interaction homme-machine réalisée sous forme d'une interrogation en langage naturel par l'intermède de l'interface. Conforme à la Figure 3 les systèmes de QR mettent en place des stratégies issues de cette discipline, Interaction Homme-Machine :

- les divers modèles d'utilisateurs ;
- plusieurs façons de présentation des réponses ;
- les interactions.

2.4.3. Recherche d'informations

La recherche d'informations (RI ou *Information Retrieval* en anglais) est la science qui étudie la manière de retrouver une information dans des documents ou des bases de données, avec le but de répondre à une question précise.

Le Vocabulaire de la documentation ([BOULOGNE, 2004]) présente la recherche d'informations différemment de recherche de l'information :

- recherche d'information : « Ensemble des méthodes, procédures et techniques permettant, en fonction de critères de recherche propres à l'usage, de sélectionner l'information dans un ou plusieurs fonds de documents plus ou moins structurés ».
- recherche de l'information : « Ensemble des méthodes, procédures et techniques ayant pour objet d'extraire d'un document ou d'un ensemble de documents les informations pertinentes ».

L'origine de la recherche d'informations est liée avec l'apparition des premiers ordinateurs. À ce moment-là est apparue l'idée d'automatiser la recherche

d'informations dans les bibliothèques. La bibliothéconomie présente les documents dans le but de récupérer des informations par la construction d'index. Pour réaliser ces recherches, il faut créer une nomenclature permettant de décrire l'ensemble des documents et, pour chaque document, sélectionner un ensemble de mots-clés. L'indexation manuelle est un processus lent qui ne garantit pas de bons résultats. À la base de cette affirmation se trouvent les connaissances insuffisantes du bibliothécaire pour traduire une question, les problèmes de synonymie ou les descripteurs pas précis. Dans ce contexte a été introduite l'extraction automatique d'un texte dans un ensemble des descripteurs.

Le Centre de Documentation et d'Information de Montpellier ([CREPS]) identifie les étapes de la recherche d'informations :

- la délimitation du sujet : comprendre la requête et délimiter la nature du travail ;
- la stratégie de recherche : en fonction du type d'information (données chiffrées, bibliographie...), de documents (articles, livres, encyclopédies...) on détermine les outils de recherche (moteur de recherche, logiciel documentaire de médiathèque...) ;
- l'interrogation des outils de recherche : en connaissant le langage documentaire et la syntaxe de recherche on construire une équation de recherche formée de mots clés et opérateurs booléens ;
- la sélection de documents : utilisant des critères de pertinence, fiabilité, validité ou actualité on sélectionne de documents ;
- le prélèvement d'information : identification et citation des sources, prise de notes ;
- le traitement de l'information : reformulation des informations trouvées et restitution sous la forme demandée.

L'article [NIE, 2008] introduit les relations du RI avec autres domaines. Concernant les systèmes des QR on voit qu'il y a des tentatives de rapproche vers RI, mais cela s'avère très difficile. La principale différence est qu'un système QR permet de répondre aux questions spécifiques à un petit domaine, fournissant une réponse directe, tandis que les systèmes RI identifie les documents nécessaires à trouver les réponses directes à la question. On pourra diminuer la distance entre la RI et la QR par l'identification du passage qui contient la réponse au lieu de fournir le document complet.

Ensuite, on met en évidence quelques similitudes entre RI et QR :

∞ la reformulation des requêtes

Cette étape consiste à considérer une requête comme un ensemble des mots-clés. On balaye les documents séquentiellement et si on trouve les mots-clés, alors il est sélectionné comme réponse.

∞ l'analyse de documents

Dans les systèmes RI et QR on réalise une opération qui d'indexation des documents par l'association de chaque document textuel à un ensemble de mots – clés. Cette structure permet à retrouver très rapidement les documents incluant des mots demandés.

∞ la mesure de pertinence

La pertinence est la notion centrale dans la RI. Article [NIE, 2008] identifie plusieurs définitions pour la pertinence : «

- *la correspondance entre un document et une requête, une mesure d'informativité du document à la requête;*
- *un degré de relation (chevauchement, relativité, ...) entre le document et la requête;*
- *un degré de la surprise qu'apporte un document, qui a un rapport avec le besoin de l'utilisateur;*
- *une mesure d'utilité du document pour l'utilisateur;* »

Ces définitions sont aussi vagues à cause du besoin très varié de l'utilisateur et des critères très différents pour juger si un document est pertinent. « *La pertinence n'est pas seulement une relation isolée entre un document et une requête, mais elle fait appel aussi au contexte de jugement.* » ([NIE, 2008]) Enfin, l'utilisateur est le seul à savoir exactement ce qu'il cherche et donc seulement il est en mesure de juger la pertinence des informations retournées.

2.5. Conclusion

Dans ce chapitre nous avons abordé de manière générale les systèmes de question-réponse, connus aussi avec le nom de *systèmes d'interrogation en langage naturel*. Un système de question-réponse utilise une phrase interrogative en entrée et en recherchant dans une base de connaissances il doit fournir une réponse courte et précise à cette question.

Pour délimiter le sujet, en consultant les articles [GIONEA, 2008] et [MORICEAU, 2009], nous avons parlé des domaines avec lesquels ils interagissent : intelligence artificielle, traitement automatique du langage naturel, interface homme-machine et recherche d'informations. Ayant à la base le document [YVON, 2007] nous avons décrit les niveaux de traitement du langage naturel : phonétique, lexical, syntaxique, sémantique, pragmatique et les principales applications existantes aujourd'hui.

Ensuite nous avons présenté l'évolution des systèmes QR au cours des années, en utilisant les articles [GENTHON, 2004] et [VASILESCU, 1996]. Si au début les chercheurs ont été fascinés de l'idée de construire des systèmes capables à répondre aux questions appartenant aux domaines restreints, en présente l'accès facile à l'information a déterminé une croissance d'intérêt pour les domaines ouverts. Tous ont comme objectif de passer le test Turing, l'idéale de l'intelligence artificielle.

Dans le chapitre suivant, nous abordons les systèmes de question-réponse du point de vue de leurs développements. Nous présentons l'architecture des systèmes QR en trois étapes : analyse de la question, recherche des documents et extraction des réponses. Ensuite nous décrirons les modalités d'évaluer les systèmes de question-réponse. À la fin, on passe en revue les principaux outils qui peuvent être utilisés, en argumentant notre choix d'outil d'implémentation pour le système d'interrogation en langage naturel pour le logiciel EXPESURF.

Chapitre 3

Développement d'un système de question-réponse

3.1. Introduction

Ce chapitre présente une architecture générale des systèmes des question réponses. Au début, nous parcourons les étapes du développement des systèmes QR: analyse de la question, recherche des documents et extraction des réponses. Ensuite, la section 3.4. présente les campagnes d'évaluation et les manières d'évaluer un système de QR. La section 3.5. identifie les principaux outils qui peuvent être utilisés dans le développement d'un système de question-réponse, tant en termes des bases de données et d'implémentation. A la fin nous argumentons notre choix d'outil d'implémentation pour le système d'interrogation en langage naturel pour le logiciel EXPESURF. Nous concluons ce chapitre par un passage en revue des articles utilisés et de notre contribution.

La suite de notre introduction présente les méthodes utilisées pour le développement d'un système de question-réponse.

Si au début des années 1960 les chercheurs ont été fascinés par l'idée de construire des systèmes capables des répondre aux questions sur des domaines restreints (closed domains), à présent le développement de l'Internet, de l'acquisition de l'information, des techniques de traitement en langage naturel et de la demande d'accès facile à l'information a déterminé l'intérêt pour les systèmes qui offre des réponses dans les domaines ouverts (open domain).

Pour construire des systèmes de question-réponse, il existe deux méthodes :

∞ La méthode type *shallow*

Cette méthode est basée sur des mots clés. Dans cette méthode on utilise des mots clés pour trouver de propositions dans les documents qui peuvent contenir des réponses valides aux questions. Ces réponses sont analysées approfondies pour établir s'ils sont bons ou pas. Cette méthode peut être utilisée avec succès pour les questions courtes et pour les domaines fermés.

∞ La méthode de type *deep*

Cette méthode implique une analyse plus compliquée : un traitement syntaxique, sémantique et contextuel. Il existe une série des méthodes qui peuvent être encadrées dans cette catégorie : *abduction*, *named-entity recognition*, *relation detection* etc. Cette méthode peut être utilisée pour les questions plus longues et pour les domaines ouverts.

Le choix de la méthode dépend de la complexité de la question et du degré de performance du système. Mais, il est clair que les systèmes de la deuxième catégorie sont supérieurs en comparaison avec la première.

3.2. Architecture

Dans cette section, nous examinons le fonctionnement des systèmes de question-réponse, en nous demandant comment ils arrivent à comprendre les textes en utilisant une méthode de développement de type *deep*.

Un système de question-réponse prend une question en entrée et en recherchant dans une base de connaissances il doit fournir une réponse courte et précise à cette question. Une base de connaissances est composée d'un certain nombre d'informations jugées utiles qui peut être représenté sous forme de données structurées (on parle alors de Database QA) ou de texte hétérogène comme pages web, documents multimédias (Text based QA, Web based QA).

La suite de cette section présente de façon générale l'architecture d'un moteur de recherche sur le Web, en consultant les documents [KATZ, 1997], [MORICEAU, 2009], [EL AYARI, 2007] et en suivant l'article [Wiki, fr]. Nous illustrons d'exemples tirés du domaine EXPESURF.

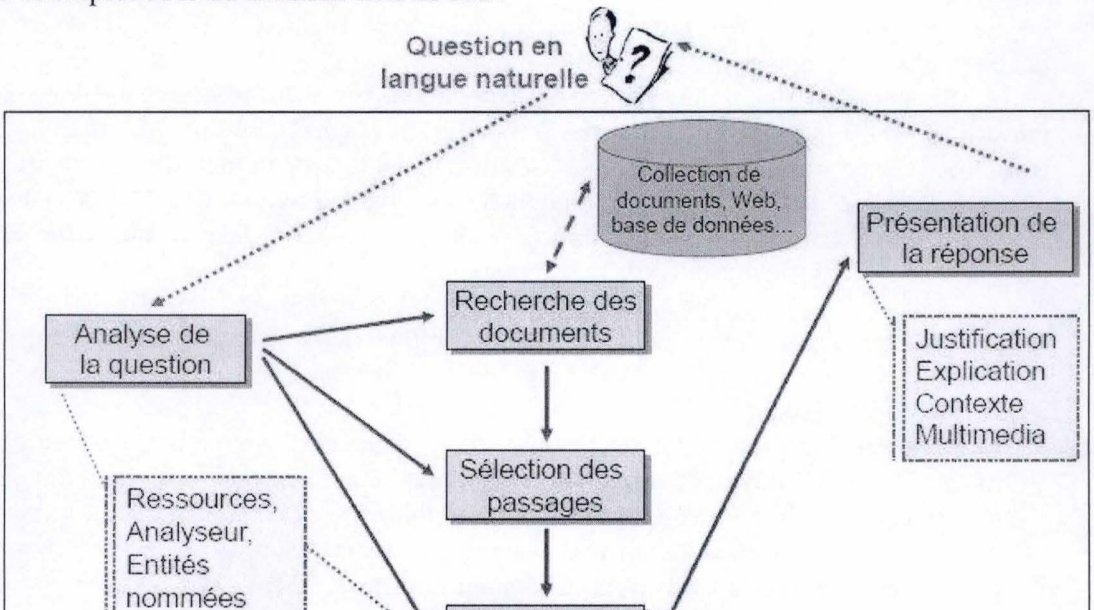


Figure 8 : L'architecture de systèmes de question-réponse
(source [MORICEAU, 2009], pag. 8)

Le traitement effectué par un SQR, présenté dans la Figure 8, se réalise dans une séquence des trois étapes : analyse de question, recherche des documents et extraction des réponses. Les informations principales se trouvent dans la question, d'où on doit extraire le plus d'éléments pertinents possible.

3.2.1. Analyse de question

Le module d'analyse des questions est une étape essentielle dans la réalisation des systèmes question-réponse, permettant des récupérer les informations essentielles pour identifier la réponse. Pour obtenir des réponses correctes, il faut réaliser une analyse plus détaillée de la question. Ce module transforme les questions posées en langage naturel en une interrogation pour le moteur d'acquisition de documents.

Cette étape vient à identifier un certain nombre d'éléments importants pour la récupération de la réponse:

- le type de la question
  Le type de la question est une caractéristique qui est utilisée pour déterminer la stratégie de traitement de la question. Nous présentons dans la figure suivante les types de question du domaine de traitement de matériaux :

Question factuelles	« Dans quel milieu sont traitées les surfaces qui contiennent Zinc ? »
Questions booléennes	« Cuivre est un matériau ? »
Définitions	« Que signifie un matériau ? »
Causes / Conséquences	« Pourquoi le FeAl est-t-il brillant ? »
Procédures	« Comment résoudre l'incompatibilité Cuivre – Fer ? »
Listes	« Citer 3 propriétés du matériau Cuivre. »
Requêtes évaluatives / comparatives	« Quel est le matériau le plus acide ? »
Opinions	« Que pensez-vous du matériau Zinc ? »

Figure 9 : Les types des requêtes

Dans notre application d'interrogation de la base de données pour le logiciel EXPESURF, présentée dans le chapitre 4, nous utilisons seulement de question de type : listes, requêtes comparatives, requête évaluative et procédures. Même si le logiciel EXPESURF est un système expert dans le domaine de surface engineering, sa base de données ne contient pas suffisamment informations pour répondre aux questions factuelles, définitions, causes / conséquences et opinions.

- le type de la réponse attendue
  En fonction du type de l'objet question ou de type de la phrase attendue on va avoir un certain type de réponse

Personne	« Qui ... », « Quel ministre ... »
Organisation	« Qui ... », « Quelle compagnie ... »
Lieu	« Où ... », « Dans quel région ... »
Date	« Quand ... », « En quelle année ... »

Figure 10 : Les types d'objets

(source [Wiki, fr])

Dans le logiciel EXPESURF, nous n'avons pas de personnes, d'organisations, de dates ou de lieux, mais seulement de matériaux, propriétés, procédées, critères et de tests.

Explication	« Pourquoi ... », « Pour quelle raison ... »
Procedure	« Comment ... », « Quelles sont les étapes pour ... »

Figure 11 : Les types de phrases
(source [Wiki, fr])

- le focus de la question
  Quand on parle de focus d'une question on fait référence à la propriété ou l'entité recherche par la question.

Exemple tiré du domaine EXPESURF:
Quelle **propriété** a le matériau « Cuivre » ?

- le thème de la question
  Le thème de la question représente l'objet sur lequel se porte la question.

Exemple tiré du domaine EXPESURF:
Quelle propriété a le matériau « **Cuivre** » ?

3.2.2. Recherche des documents

À partir des données reçues par le module de l'analyse de question, on cherche dans la collection des documents, les articles spécifiques pour la question posée par l'utilisateur. Pour réaliser ce but, on parcourt plusieurs étapes, suivant la Figure 12.

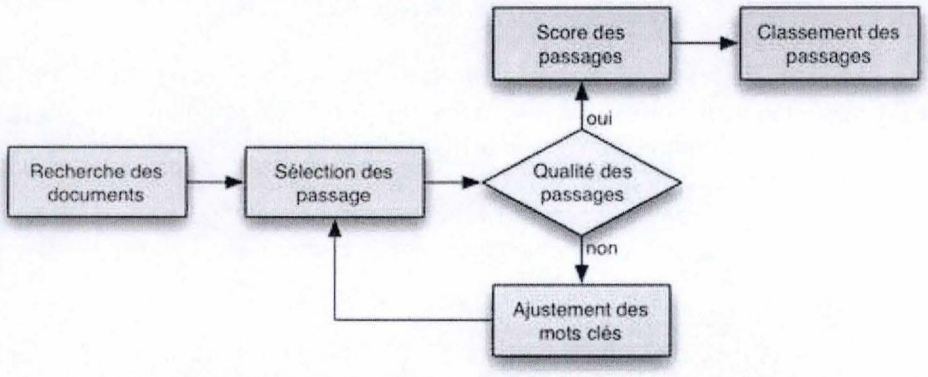


Figure 12 : Recherche des documents dans un système de question-réponse
(source [MORICEAU, 2009], pag. 13)

La première étape dans la recherche des documents est une recherche basée sur l'extraction de mots clés de la question. La recherche peut en suite être réalisée en consultant une base de données ou le moteur de recherche Google.

Exemple tiré du domaine EXPESURF:

Question	Mots clés
Quelles sont les propriétés du matériau Cuivre ?	propriétés, matériau, Cuivre

Figure 13 : Exemple d'extraction des mots clés

Avec les mots clés obtenus par étape antérieure on sélectionne les passages correspondants. Pour réaliser une première sélectionne des passages on utilise le type de la réponse attendue, les mots-clés extraits de la question et ses entités nommées. Ensuite, on estime la qualité des passages sélectionner. Dans le cas où on n'obtient pas une qualité désirée, on va réajuster les mots-clés et on recommence l'étape de la sélection des passages. Après on analyse le nombre de passages obtenus. Si l'on obtient trop, on restreindra le nombre de mots-clés, si au contraire il n'y en a pas assez, on étendra la requête.

En parcourant la Figure 12, l'étape suivante de la sélectionne des passages est de leur attribuer un score et de les classer. On va introduire deux notions : de passage candidat et de réponse candidate. Le *passage candidat* est un texte (d'une phrase à un document entier) sélectionné par le moteur de recherche. Selon l'index utilisé il peut contenir ou pas de réponses candidates. Généralement, ils sont classés à l'aide d'un score attribué par le moteur de recherche. La *réponse candidate* est un mot ou un groupe de mots qui possède le même type que le type de réponse attendue.

Exemple tiré du domaine EXPESURF:

Question	Quel matériau a la propriété dureté plus petit que 5 ? -> **Type matériau**
Passages candidats et Réponses candidates	Le matériau **Cuivre** est un matériau de dureté 3. Le laiton est un alliage formé par le matériau **Cuivre** et le matériau **Zinc**.

Figure 14 : Exemple de passage candidat et réponse candidate

3.2.3. Extraction des réponses

En fin, le troisième module d'extraction de la réponse utilise les articles retournés du module d'acquisition des documents et extrait une réponse succincte qui constitue la réponse cherchée par l'utilisateur. Le processus d'extraction dépend du type de réponse attendue : quand la réponse a un certain type d'entité de type nom, le module identifie ces entités dans chaque proposition extraite. Au cas contraire si le type de réponse n'est pas un nom d'entité, le processus d'extraction utilise les principes de reconnaissance du focus.

On présente ci-dessous quelques techniques pour l'extraction de réponses.

3.2.3.1. Patrons d'extraction

Le patron d'exécution est une technique d'extraction de réponses, héritée des techniques de recherche d'informations. Cette technique recherche une séquence calque modélisant la réponse, où l'élément cherché correspond à une variable.

Exemple tiré du domaine EXPESURF:

Question	Patron
Quelles sont les propriétés du matériau Cuivre ?	X est la propriété du matériau Cuivre

Figure 15 : Exemple de patron d'extraction

Aujourd'hui un problème pour les systèmes de question-réponse est posé par la variabilité linguistique. Pour le résoudre, nous utilisons pour la phase d'extraction des réponses plusieurs variantes pour chaque calque. Nous identifions les types de variations et les exemples suivants pris du [Wiki, fr] :

- **Variation morphologique** :
  « Où se trouve la *capitale de l'Europe* » ou
  « Où se trouve la *capitale européenne* ? »
- **Variation lexicale** :
  « Comment s'appelle la reine de *Hollande* » ou
  « Comment s'appelle la reine des *Pays-bas* ? »
- **Variation syntaxique** :
  « Moscou compte 9 millions d'habitants » ou
  « Les 9 millions d'habitants de Moscou »
- **Variation sémantique** :
  « Comment Adolf Hitler est-il *mort* ? » où la réponse peut être
  « Adolf Hitler *s'est suicidé* »

Le monde des traitements de surface des matériaux est un domaine très précis. Nous n'avons pas des synonymes pour le matériau, les procédés, les propriétés, etc. Les seules variations rencontrées sont celles syntaxiques. Par exemple nous pouvons utiliser : « Le matériau Cuivre possède des propriétés. » ou « Les propriétés du matériau Cuivre ».

Mais pour les questions appartenant à un domaine ouvert cette méthode d'extraction à la main des patrons est très couteuse et prend beaucoup de temps. Une solution à ce problème est l'utilisation des méthodes d'apprentissage qui réalise une extraction automatique.

L'extraction automatique de réponses est réalisée en plusieurs étapes :

- le choix d'un patron ;
- le remplacement d'un élément du patron par une variable ;
- l'extraction du corpus d'un ensemble d'éléments pouvant instancier le patron.

Exemple tiré du domaine EXPESURF:

Patron de base		Patrons acquis
<test>X</test> évalué <propriété>Y</propriété>		X<évaluer>Y
X a évalué Y	évaluer(X,Y)	<évaluation de>Y<par>X

Y a été évalué par X l'évaluation de Y par X	évalué(X,Y) évaluation(X,Y)	

Figure 16 : Exemple de patron de base et patrons acquis

3.2.3.2. Score et critères

Une autre technique utilisée pour l'extraction des réponses est d'associer à chaque réponse candidate un score. Pour calculer le score, nous utilisons quatre critères différents, cités du [Wiki, fr]: «

- **Bon contexte global** : on évalue la pertinence du passage qui contient la réponse candidate. Pour cela on se base sur :
  - le nombre de mots-clés présent dans le passage,
  - le nombre de mots communs à la question et au passage,
  - le classement du moteur de recherche pour le passage,
- **Bon contexte local** : on évalue l'adéquation du passage par rapport à la question :
  - Distance moyenne entre la réponse candidate et les mots-clés présents dans le passage,
  - Nombre de mots de la réponse candidate qui ne sont pas des mots-clés de la question,
- **Type sémantique correct** : on vérifie si le type de la réponse candidate est soit le même soit un sous-type du type de réponse attendue.
- **Redondance** : présence de la réponse dans le plus possible de passages sélectionnés. »

3.2.3.3. Relations syntaxiques

Une analyse syntaxique de la question et des passages candidats donne une amélioration de l'extraction des réponses. Pour aller plus loin, on introduit une analyse sémantique en construisant des arbres de dépendances.

Ensuite nous présentons un exemple provenu de l'article [Wiki, fr]. Si nous considérons la question « Quel métal a le plus haut point de fusion ? », par une analyse syntaxique nous obtenons les suivantes relations :

Question	**Relations**
Quel métal a le plus haut point de fusion ?	[X, métal], [point de fusion, maxime]
Réponse candidate	**Relations**
Tungstène est le matériau avec le plus haut point de fusion.	[Tungstène, métal] [point de fusion, maxime]

Figure 17 : Relations syntaxique pour la question « Quel métal a le plus haut point de fusion ? »

Si nous analysons la question du point de vue de la sémantique, nous observons que l'objet cherché est « METAL » de type « métal ». Après la recherche dans les documents nous obtenons le passage candidat « Tungstène est un métal très dur et a le plus haut point de fusion. » qui contient notre réponse Tungstène.

La Figure 18 nous présentons l'arbre de dépendances.

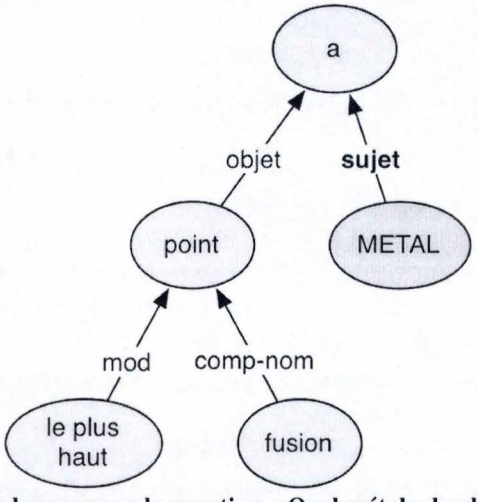


Figure 18 : Arbre de dépendances pour la question « Quel métal a le plus haut point de fusion ? » (source [Wiki, fr])

Dans la Figure 19 sont présentés les relations syntaxiques à parti de l'arbre de dépendances.

Question	Passage
<METAL, a, sujet>	<tungstène, métal, pred>
<point, a, objet>	<tungstène, a, sujet>
<fusion, point, comp-nom>	<point, a, objet>
<le plus haut, point, mod>	...

Figure 19 : Relations syntaxiques à parti de l'arbre de dépendances (source [Wiki, fr])

Le déroulement de ces étapes est présenté dans la figure suivante :

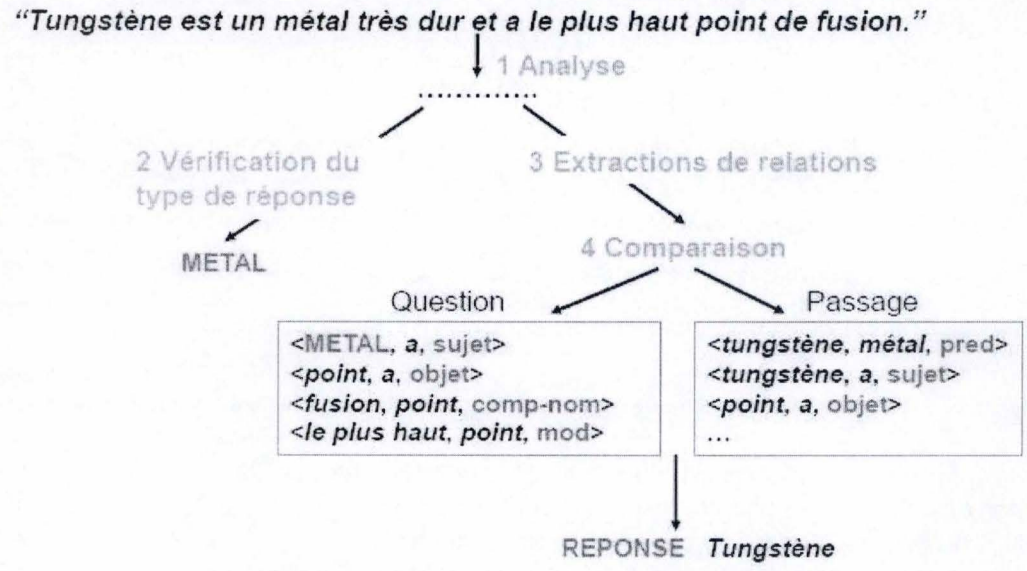


Figure 20 : Extraction des réponses utilisant de relations syntaxiques

(source [MORICEAU,2010, page 28])

3.2.3.4. Utilisation de la logique

Une autre technique de l'extraction de réponses est l'utilisation de la logique. Par l'ajout des prédicats et des règles de subsomption (la relation « est impliqué par » en logique classique, ou encore « contient » en logique ensembliste) on va convertir les passages en formules logiques et la question en but. L'objectif est de prouver le but à partir du passage.

Par exemple pour la proposition « Le matériau Fer a la propriété dureté. », nous pouvons la représenter dans la logique de propositions sur la façon suivante :

materiau (FER)
est-propriete (FER, DURETE)

3.3. Difficultés rencontrées

Même si l'Internet est un milieu avec des connaissances des tous les domaines, à trouver une réponse à une question est une tâche difficile. On distingue quelques types des difficultés qui peuvent apparaître :

∞ formulation correcte de requêtes

La transformation d'une question en langage naturel à une requête pour un moteur de recherche est une tâche difficile. Si la question est trop générale seront extraits un nombre des documents très grand. En outre, il est possible que les passages extraits ne contiennent pas exactement la réponse à la question de l'utilisateur. Si on extrait trop de documents, le temps de traitement sera augmenté. En autre cas, si l'ensemble de mots-clés est trop petit, il est possible de ne pas trouver d'article qui réponde à la question. En conséquence, il faut avoir une question bien formulée pour extraire seulement les documents qui contiennent l'information utile.

∞ bruit

Même si nous avons trouvé le bon ensemble de mots-clés pour réaliser une interrogation qui est en mesure de retourner les articles avec des informations utiles, le moteur de recherche peut retourner un grand nombre d'articles qui ne répondent pas à la question de l'utilisateur. Par exemple, si la question « Qui a été le premier homme dans l'espace?", le module d'acquisition de l'information doit recevoir les mots clés « le premier homme dans l'espace ». Mais le moteur de recherche est susceptible d'avoir des pages indexées pour « le premier touriste de l'espace» ou «l'homme le plus âgé dans l'espace» et il va les retourner avec des articles contenant la réponse correcte.

∞ informations fausses

Même si la question est bien formulée et le moteur d'acquisition retourne des articles qui correspondent à la question soumise, il est possible que certains de ces articles contiennent des informations erronées. Dans ce cas, le système n'a aucun moyen de savoir quelles sont les réponses correctes ou incorrectes.

∞ ressources limitées

Quand on construit un système pour répondre aux questions, il faut tenir compte de limitations imposées par le travail avec de grandes quantités d'informations. Il n'est pas indiqué d'envoyer au système un ensemble de chaînes de

mots trop grand pour les interrogations. Bien que les moteurs de recherche actuels soient assez rapides et le retour des réponses soit succinct, la recherche dans les longues listes d'articles prend beaucoup de temps et l'utilisateur du système n'est pas prêt à attendre quelques minutes pour que le système leur donne une réponse.

3.4. Évaluation

La réalisation d'une campagne d'évaluation offre la possibilité de faire collaborer des chercheurs de domaines différents, de confronter les différentes approches sur un même problème, de favoriser les transferts technologiques et de faire avancer l'état de l'art.

Les enjeux d'évaluation des systèmes de question-réponse ont pris une importance plus forte avec l'apparition de la huitième édition du TREC[8], qui a été consacrée en 1999 à une piste des systèmes QR spécifiques. Comme indiqué dans [EL-BEZE, 2006], depuis TREC-8 l'évaluation des systèmes de question-réponse a été organisée chaque année variant le nombre de questions posées (entre 200 et 700), la taille de la réponse, le nombre de réponses autorisées (entre 1 et 5), la taille de la collection de documents mise à disposition des participants.

Aujourd'hui, on distingue essentiellement cinq campagnes que sont TREC (depuis 1999, anglais), CLEF[9] (depuis 2003, multilingue), EQUER[10] (depuis 2004, français), NTCIR[11] (depuis 2002, japonais et chinois), QUAERO[12] (depuis 2008, français, anglais). Les systèmes sont évalués autant à partir de domaine ouvert que de collections fermées (en général, des articles de journaux).

Tenant compte qu'il y a plusieurs façons de répondre en langage naturel à une question en langage naturel, on ne pose la question : comment évaluer ces systèmes ? Comme indiqué dans [EL-BEZE, 2006] on devrait être capable de juger si « *la réponse apportée est juste, concise, complète, appropriée (en fonction d'un contexte particulier), rapide, détaillée, approfondie, étayée* ».

Il y a deux principales manières d'évaluer les systèmes de question-réponse. Le premier type de jugement est la mesure automatique adoptée par la Moyenne des Rangs Réciproques. Nous appellerons ici MIR pour *Moyenne de l'Inverse du Rang*[13]. Cette moyenne est définie de la façon suivante :

$$MIR = \frac{1}{N_q} \times \sum_{q=1}^{Nq} \frac{1}{R_q}$$

Équation 2 : Definition du Moyenne de l'Inverse du Rang

où : Nq est le nombre de questions ;
Rq est le rang auquel le système a classé la réponse attendue pour la question q.

[8] TREC (Text REtrieval Conference) : http ://trec.nist.gov/
[9] CLEF (Cross Language Evaluation Forum) : http ://clef.isti.cnr.it/
[10] EQUER (Évaluation en Question Réponse) : http ://www.technolangue.net/article195.html/
[11] NTCIR (NII Test Collection for IR Systems) : http://research.nii.ac.jp/ntcir/outline/prop-en.html
[12] QUAORO : http://www.quaero.org/
[13] MIR ou en anglais *Mean Reciprocal Rank* (MRR).

Le second moyen d'évaluation d'une réponse est le *jugement humain*. On se base alors sur la correction ou l'exactitude de la réponse ainsi que sa justification. L'article [TUFIS, 2008] mentionne qu'une réponse est correctement évaluée pas seulement par son contenu, mais aussi par l'identification correcte du passage candidat. Il présente le jugement humain selon quatre critères :

- correcte : la réponse est correct, le texte d'où il est extrait est correct ;
- incorrecte : la réponse n'est pas correct indifférent si le texte d'où il a été extrait est correct ou pas ;
- non exacte : la réponse est partiel correcte, trop longe ou trop bref, mais le texte d'origine est correct ;
- non justifiée : la réponse est correcte, mais le texte d'origine est incorrect.

Les articles [MORICEAU, 2009] et [NIE, 2008] présentent deux métriques importantes : le rappel et la précision. Ces mesures impliquent l'existence d'un ensemble de documents et la connaissance des documents pertinents à contenir les réponses à la question.

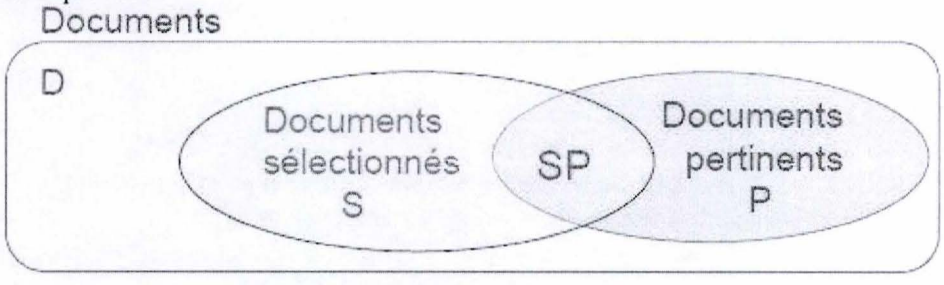


Figure 21 : Diagramme des documents

(source [MORICEAU, 2009])

L*e rappel* mesure la proportion des documents pertinents sélectionnés parmi l'ensemble des documents pertinents :

$$\frac{\text{documents pertinents sélectionnés } SP}{\text{documents pertinents } P}$$

Figure 22 : Définition du rappel

(source [MORICEAU, 2009])

On observe que le rappel est meilleur s'il y a moins de documents pertinents qui n'ont pas été sélectionnés.

La précision mesure la proportion de documents pertinents sélectionnés parmi tous les documents sélectionnés :

$$\frac{\text{documents pertinents sélectionnés } SP}{\text{documents sélectionnés } S}$$

Figure 23 : Définition de la précision

(source [MORICEAU, 2009])

Selon la Figure 23, la précision est plus grande s'il y a moins de mauvais documents sélectionnés.

Idéalement, on voulait qu'un système donne de bons taux de précision et de rappel en même temps. Mais les deux métriques ne sont pas indépendantes : quand l'une augmente, l'autre diminue. On présente graphiquement la variation de la dépendance entre précision et rappel dans la Figure 24 :

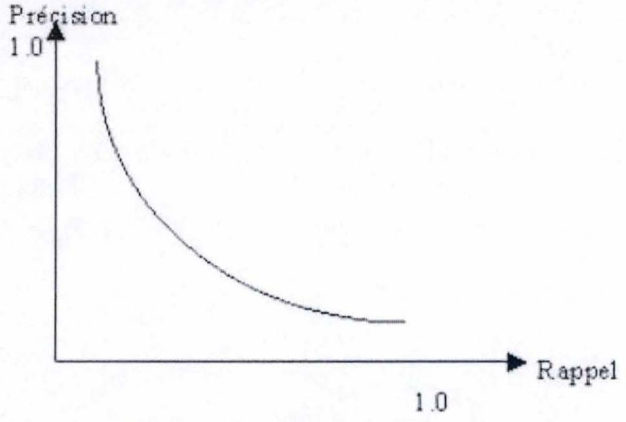


Figure 24 : Courbe de dépendance entre le rappel et la précision

(source [NIE, 2008])

À partir des figures présentées ci-dessus, on conclut qu'une augmentation du rappel entraîne une baisse de la précision et vice versa.

L'article [NIE, 2008] décrit comment comparer deux systèmes utilisant ces métriques. Le système est meilleur si sa courbe de dépendance entre le rappel et la précision est située plus haut que l'autre. Dans le cas où les deux courbes se croisent, il est difficile de mesurer la performance. Pour résoudre ce problème, nous utilisons *la précision moyenne.* « *La précision moyenne est une moyenne de précision sur un ensemble de points de rappel.* » ([NIE, 2008]). TREC utilise dans ces évaluations 11 points de rappel : de 0 à 1 par pas de 0.1.

Dans ce sous-chapitre, nous avons présenté les méthodes d'évaluation des systèmes de question-réponse. Au début, nous avons parlé des campagnes d'évaluation existantes. Comme l'article [EL AYARI, 2007] indique, celles-là permettent une évaluation globale des systèmes entre eux en imposant des points d'étalons de référence, des processus bien définis et des méthodes de comparaisons. En même temps, elles mesurent chaque année l'avancement de la technologie de ces systèmes, permettant de stimuler la recherche.

De l'autre côté, nous avons présenté l'évaluation par le jugement humain, car seulement l'utilisateur est à l'origine du besoin en informatique. Ensuite, nous avons introduit des métriques qui ont pour but de permettre la comparaison des modèles entre eux et l'analyse fine des performances de système de QR.

3.5. Outils pour les systèmes de question-réponse

Les systèmes de question-réponse utilisent dans leur implémentation une base de données contenant des informations, une interface en langage naturel qui permet la communication entre l'homme et ordinateur et un logiciel qui effectue le traitement du langage naturel. Certains systèmes qui réalisent une analyse plus compliquée disposent des outils de traitement du langage naturel.

Dans cette section, nous identifions les principaux outils qui peuvent être utilisés dans le développement d'un système de question-réponse, tant en termes des bases de données et d'implémentation.

3.5.1. Outils de base des données

La base de données est un ensemble des informations stockées de manière organisée. Laurent Audibert dans la livre « Base de données – de la modélisation au SQL », définie la base de données comme « *un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, représentant des informations du monde réel et pouvant être interrogées et mises à jour par une communauté d'utilisateurs.* »

En ce qui concerne l'utilité d'une base de données, elle permet la centralisation, la coordination, l'intégration et la diffusion de l'information archivée. Les utilisateurs peuvent consulter, mettre à jour ou effacer des données, s'ils disposent des droits pour effectuer ces opérations. L'avantage majeur de l'utilisation de bases de données est la possibilité d'être accédées par plusieurs utilisateurs simultanément.

La majorité des bases de données utilisées sur l'ordinateur sont des bases de données relationnelles qui supportent l'utilisation de SQL. Un *système de gestion de base de données (SGBD)* est un ensemble des programmes et des langages de commande qui permettent l'accès (interrogations, mises à jour, calculs, extractions) et la gestion de base de données. Parmi les caractéristiques d'un SGBD, on identifie :

- l'indépendance physique ;
- l'indépendance logique ;
- la rapidité d'accès ;
- l'administration centralisée ;
- la limitation de la redondance ;
- la vérification de l'intégrité ;
- le partage des données ;
- la sécurité des données.

Cette sous-section présente le langage SQL et les plus importants systèmes de gestion de base de données utilisés par des systèmes de question-réponse : MySQL, PostgreSQL et Oracle.

3.5.1.1. Le langage SQL

Le langage SQL (Structured Query Language) est un langage de programmation spécifique pour la communication avec les bases de données. Audibert précise dans sa livre «Base de données de la modélisation au SQL » que le langage SQL a le succès d'aujourd'hui grâce à *« sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution »*. [AUDIBERT, 2009]

Les principales commandes du langage SQL se résument à cinq opérations de manipulation de base des données :

- la création d'une table ;
- la création de tables obtenues par jointure ;
- la lecture d'une sélection d'une table ;
- l'effacement d'une table ;
- l'insertion de nouvelle ligne dans une table ;
- l'effacement des lignes dans une table ;
- la modification des lignes dans une table.

Quels sont les avantages du langage SQL ?

SQL n'est pas un langage utilisé seulement par les distributeurs des bases de données individuelles. Au contraire, presque toutes les bases de données importantes, comme MySQL, Oracle, PostgreSQL, utilisent ce langage. Donc son apprentissage ne permet d'interagir avec les bases des données le plus connues. Un autre avantage est que SQL est facile à apprendre : il contient moins d'instructions qui sont composées de mots descriptifs (en anglais). Malgré à sa simplicité, SQL est un langage puissant qui peut effectuer des opérations complexes sur des bases de données sophistiquées.

3.5.1.2. MySQL

MySQL[14] est un système de gestion de base de données parmi le plus utilisé au monde, en concurrence avec Oracle, Microsoft SQL Server ou PostgreSQL. Il est utilisé autant par le grand public que par des professionnels, étant disponible sous une licence libre, mais il existe aussi des licences commerciales.

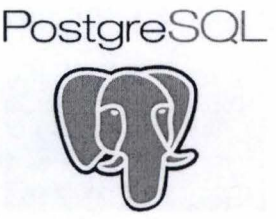
Le serveur MySQL contrôle l'accès aux informations de la base de données en garantissant que plusieurs utilisateurs peuvent la manipuler simultanément. Donc il est un serveur multithreading et multiutilisateur.

Le système MySQL offre les suivantes avantages à ses utilisateurs:

- une grande vitesse d'accès aux données;
- la gratuité;
- une sécurité élevée;
- la portabilité.

[14] MySQL : www.mysql.com

3.5.1.3. PostgreSQL



PostgreSQL[15] est un système relationnel de gestion de la base de données compatible avec SQL. L'une des principales qualités est d'être un logiciel libre.

PostgreSQL fonctionne selon une architecture client/serveur: de coté serveur il y a une application qui est capable de traiter les requêtes des clients et de coté client il est imposée une installation sur les machines pour interroger le serveur de bases de données à l'aide de requêtes SQL. Est un modèle client – serveur relatif simple qui utilise un processus pour chaque requête de l'utilisateur. ([AUDIBERT, 2009])

Ensuite on présente quelques caractéristiques du système PostgreSQL. L'utilisateur peut créer des types, des fonctions ou utiliser l'héritage de type, effectué directement dans le moteur de la base de données, via PL/pgSQL. PostgreSQL peut être utilisé sur les plateformes Windows et Unix et offre des interfaces disponibles pour tous les langages de programmation importante : C, Perl, Python, Java et PHP.

PostgreSQL possède quelques restrictions mineures. La grandeur d'une base de donnée et illimitée, mais la grandeur d'une table est limitée à 64 téraoctets. Chaque champ peut avoir jusqu'à 1 gigaoctet. Le nombre des lignes d'un tableau est illimité, mais le nombre des colonnes est maximum 1600. Donc, PostgreSQL est un système capable à travailler avec un volume grand de données.

Notre logiciel EXPESURF utilise comme base de données PostgreSQL et l'interface web d'administration phpPgAdmin.

3.5.1.4. Oracle

Un autre système de gestion de base de données relationnelle (SGBDR) qui utilise le langage d'interrogation SQL est Oracle Database. Il est fourni par Oracle Corporation[16], et a été développé par Larry Ellison.

L'utilisation d'Oracle présente les suivantes avantages significatifs, en effet Oracle

- est un langage ouvert qui respecte les standards liés aux langages d'accès aux données (SQL) ;
- supporte de base de données de n'importe quelle dimension ;
- supporte simultanément un grand nombre d'utilisateurs, minimisant les conflits d'accès aux données et garantissant la simultanéité des données ;
- implémente un milieu client – serveur ;
- offre la sécurité et surveille l'accès aux données et leurs utilisations ;
- est portable sur toutes les plateformes (Windows, UNIX, Macintosh).

[15] PostgreSQL : www.postgresql.org
[16] Oracle Corporation : www.oracle.com

3.5.2. Outils de développement

Aujourd'hui il y a une grande diversité des langages de programmation. Les plus utilisés en ingénierie du langage sont PROLOG, LISP, PERL et JAVA.

3.5.2.1. Prolog

Le mot Prolog[17] provient de PROgramming in LOGic. Créé par Alain Colmerauer et Philippe Roussel, il est apparu dans les années 1972. Le but des auteurs a été de « *concevoir un langage de programmation qui permettait d'utiliser l'écriture de la logique à la place de la séquence d'opérations* »[18]. Un événement de référence dans l'évolution du langage a été la conférence de l'année 1981 à Tokyo, où Prolog a été choisi par les chercheurs japonais comme un langage de base pour les ordinateurs de cinquième génération.

Un programme Prolog est une collection de faits et de règles qui sur base d'une requête permettent d'extraire les réponses existantes dans l'ensemble des faits. L'article [HARMACH et al., 2003] décrit un programme Prolog comme un ensemble de clauses appelées *clauses de Horn*[19]. « *Tout ce qui est écrit est vrai et tout ce qui n'est pas écrit est considéré comme faux par Prolog. Toute exécution d'un programme Prolog est une preuve.* » L'article parle aussi de fonctionnement d'un programme logique : « *Un programme logique est un ensemble d'axiomes ou de règles définissant des relations entre objets. Un calcul d'un programme logique est une déduction de conséquences à partir du programme. Un programme définit un ensemble de conséquences qui est sa signification. L'art de la programmation logique consiste en la construction de programmes concis et élégants qui ont la signification souhaitée. Le résultat de l' exécution d'un programme Prolog est conséquence logique des axiomes qu'il contient.*»

Le mécanisme de fonctionnement du langage Prolog este basé sur backtracking et utilise comme structure de données de base les listes. Par exemple, les propositions sont représentées sous format de listes d'atomes. Ce n'est pas un hasard, car la manipulation de listes est très bien adaptée pour l'IA. Par exemple la question « Quelles sont les propriétés du matériau Cuivre ? » va être exprimée dans la liste suivante : [Quelles, sont, les, propriétés, du, matériau, Cuivre, ?]

Ce langage est très appliqué dans le domaine d'intelligence artificielle. Maintenant il profite d'une extension de plus en plus grande, étant utilisé dans le traitement du langage naturel. La stratégie du programme pour traiter les propositions est simples : on donne une requête et Prolog réalise plusieurs pas : une lecture de la proposition, son parsage en découvrant les formes logiques, la conversion des formes logiques en clause Prolog, puis il réalise la génération et l'affichage d'une réponse et la répétition des ces pas.

---

[17] Prolog: www.swi-prolog.org/
[18] http://general.developpez.com/langages/?page=tous#prolog
[19] Clause de Horn est une clause comportant au plus un *littéral positif.*

Prolog est un langage de très haut niveau qui privilégie l'écriture et la lisibilité du programme. On énumère quelques exemples des logiciels développés en Prolog et appliqués dans le domaine d'intelligence artificielle: ELIZA[20] , CHAT[21].

3.5.2.2. LISP

Le deuxième langage le plus utilise pour l' intelligence artificielle est LISP[22] (LISt Processor). LISP est un langage de programmation inventé à la fin des années '50 qui a la syntaxe la moins restrictive parmi les langages de programmation de haut niveau.

Une particularité importante est que les programmes sont traités comme des données. En LISP tout le programme peut être considéré comme donnée d'entrée pour autres programmes ou fonctions. Comme le nom l'exprime, la liste est l'unité de base. La raison pour laquelle la liste, une structure peu spectaculaire, peut être à la base d'un domaine complexe comme intelligence artificielle est que cette structure est très générale. Elle peut contenir des nombres, mais aussi des autres listes, donnant la possibilité de représenter d'une manière uniforme les structures compliquées.

LISP est un langage bien utilisé dans le domaine d'intelligence artificielle grâce à ses caractéristiques : la facilité de travailler avec les listes, les mécanismes complexes d'évaluation et l'utilisation des macros[23].

Lisp a eu une série de successeurs, dont le langage Scheme, Haskell.

Haskell[24] est un langage de programmation fonctionnel, fondé sur le lambda-calcul et la logique combinatoire. Les caractéristiques principales du Haskell sont l'utilisation des fonctions récursives, de l'inférence de types[25], des listes en compréhension[26] et de l'évaluation paresseuse[27].

Scheme[28] est un langage dérivé du LISP et crée par MIT dans le but d'épurer le LISP. Une caractéristique du Schème est la syntaxe extrêmement simple avec un nombre très limité de mots-clés. En plus il est un langage multi-paradigme[29]. Il conserve du LISP les aspects essentiels, la flexibilité et la puissance expressive. ([SPERBER, 2007])

---

[20] ELIZA: chapitre 2.3.
[21] CHAT: chapitre 1.6.
[22] Lisp: http://www.lisp.org
[23] Un macro est une fonction qui n'évalue pas ses arguments et se comporte différemment en fonction de contexte.
[24] Haskell: http://www.haskell.org
[25] L'inférence de types est un mécanisme qui permet à un compilateur ou un interpréteur de rechercher automatiquement les types associés à des expressions, sans qu'ils soient indiqués explicitement dans le code source.
[26] Listes en compréhension sont des listes dont le contenu est défini par filtrage du contenu d'une autre liste.
[27] L'évaluation paresseuse est une technique de programmation où le programme n'exécute pas de code avant que les résultats de ce code ne soient réellement nécessaires.
[28] Scheme: http://plt-scheme.org/
[29] Langage multi-paradigme est un langage qui supporte des paradigmes multiples (oriente objet, procédurale)

Le développement du web a relancé l'intérêt pour la programmation fonctionnelle. A mon sens, parce que c'est un langage déclaratif de haut niveau qui permet de produire des pages web comme résultat du calcul de fonctions.

3.5.2.3. PERL

PERL[30] (Practical Extraction and Report Language) est un langage de programmation créé par Larry Wall en 1987. C'est un langage interprété, dans le sens où les instructions ne sont pas transformées en code exécutable. Il est utilisé pour le traitement et la manipulation des fichiers texte. S. Buraga et al. dans la livre «Programmation Web in bach et Perl» ([BURAGA, 2002]) présente les principes utilisés dans le développement du langage :

- ∞ il existe plusieurs façons de réaliser un programme en fonction du degré de connaissance du langage ;
- ∞ les choses simples sont faciles à réaliser et ces-la complexes possible à être implémentés.

Perl a à la base plusieurs langages (C et Lisp et les langages de scripts sed, AWK et shell), présentant les caractéristiques suivantes :

- ∞ modularité
  Le langage Perl peut être étendu par d'un nombre impressionnant de modules standard.

- ∞ portabilité
  Les programmes Perl peuvent s'exécuter sur n'importe quelle plateforme sans modification.

- ∞ expressivité et puissance
  Le langage manipule les données par l'intermède des mécanismes puissantes : les expressions régulées et les tableaux.

- ∞ vitesse de développement des applications
  Perl n'offre pas un compilateur classique, mais un compilateur-interpréteur. Le cycle compilation – exécution – dépannage est rapidement réalisé.

Étant un langage gratuit avec beaucoup de documentation online, Perl est utilisé pour une grande variété des applications d'administration des systèmes d'exploitation, de développement web, de réseaux, interface graphique et autres.

[30] PERL: www.perl.org

3.5.2.4. Java

Java[31] est un langage orienté objet, créé au début des années '90 par James Gosling à Sun Microsystems (maintenant filiale d'Oracle). Java est un logiciel libre qui a été développé à partir du plus populaire langage C++. Aujourd'hui Java est omniprésent sur le marché d'informatique. Il est utilisé dans le domaine médical, l'armée, les banques et le commerce, mais aussi dans les applications pour téléphone ou jeux.

Java est un langage flexible avec lequel on peut gérer facilement la sécurité, les connexions entre les systèmes distribues, les transactions ou les Web Services. La modularité et l'abstraction assurent la qualité des applications et la facilité de maintenance. ([BAUZON, 2009])

Même si dans le domaine d'intelligence artificielle Prolog et Lisp sont les plus reconnus, on retrouve Java dans beaucoup d'outils de traitement du langage naturel (Open NLP[32], Learning Based Java[33], etc.). Par exemple un système de question-réponse qui utilise comme langage de développement Java est celui décrit dans l'article [BOUCHOU, 1999] : « Une bibliothèque d'opérateurs linguistiques pour la consultation de base de données en langue naturelle ».

Son succès actuel est déterminé de « *la souplesse du langage, ses qualités en termes de développement, sa portabilité, son interopérabilité vers diverses plates-formes et la quantité de frameworks disponibles* » (selon l'article [BAUZON, 2009]). De plus, la Java Virtual Machine assure la portabilité du langage. Elle nous permet de développer des logiciels en faisant abstraction du système d'exploitation. Une autre avantage de l'utilisation du langage Java est le nombre impressionnant de librairies disponibles.

De point de vue du développement des applications de bases de données, Java est un excellent candidat grâce à ses caractéristiques: robustesse, sécurité, facile à comprendre. La communication entre les bases de données et les applications Java est réalisée par le connecteur JDBC. JDBC[34] (Java Database Connectivity) est une interface standard SQL pour accéder à des bases de données.

La programmation Java, comme toutes les méthodes de programmation, suppose l'écriture de fichiers sources qui sont compilés. Souvent il existe des erreurs de syntaxe, de logique ou seulement d'écriture. Pour faciliter leurs résolutions, nous utilisons un IDE[35], un logiciel spécialisé dans l'aide au développement d'applications. Dans le cas de Java, le plus répandue et facile à utilise est le moyen de développement Eclipse, présenté au-dessous.

---

[31] Java: http://www.oracle.com/technetwork/java/index.html

[32] OpenNLP est une machine d'apprentissage automatique pour le traitement de texte en langage naturel.
WebSite : http://incubator.apache.org/opennlp/index.html

[33] LBJ est un langage de programmation basé sur Java, orienté sur l'apprentissage machine et le traitement du langage naturel (NLP). WebSite : http://cogcomp.cs.illinois.edu/page/software_view/11

[34] JDBC pour PostgreSQL: http://jdbc.postgresql.org/

[35] Integrated Development Environment

Moyen de développement : Eclipse

Eclipse[36] est un moyen de développement open-source, un IDE qui contient les trois outils: un éditeur de texte, un compilateur et un débogueur. Créé par la Fondation Eclipse, il est utilisé pour le développement des applications Java, mais aussi C, C++, Python, Perl, PHP par l'intermède des plug-ins.

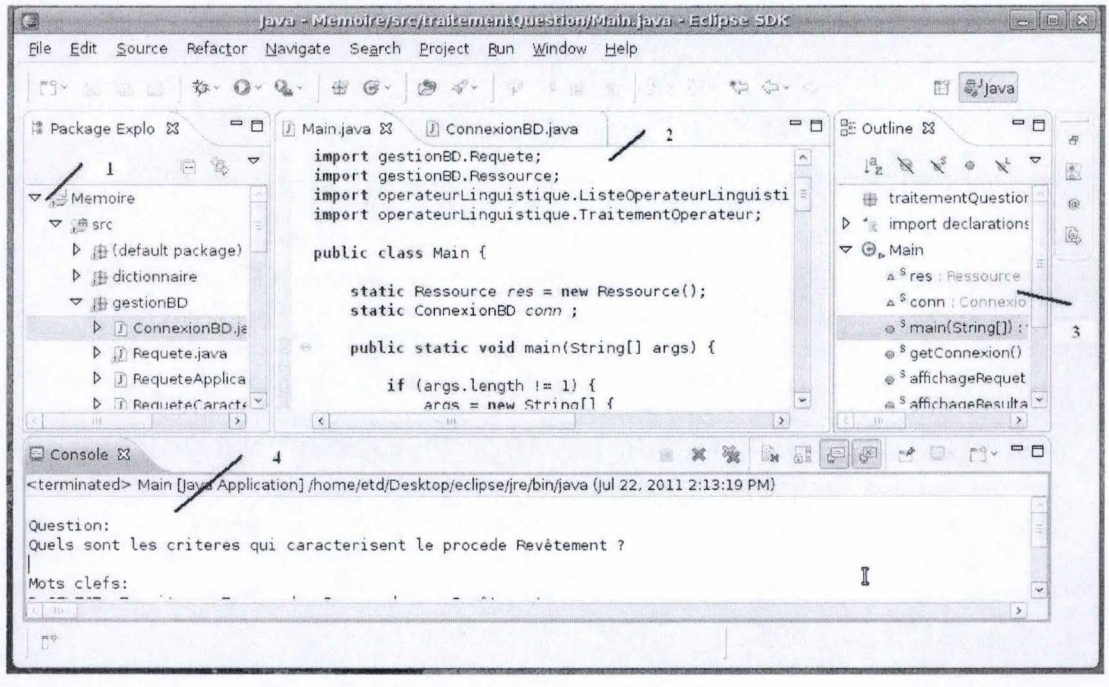


Figure 25 : L'environnement Eclipse

Comme on observe dans la Figure 25 l'environnement de travail sous Eclipse est divisé en quatre parties principales :

1 – est la partie de gauche de l'environnement qui nous permet de naviguer dans les projets et les paquètes ;

2 – est la partie centrale qui nous permet d'éditer en même temps plusieurs classes ou textes présentés en onglets différents

3 – est la partie de droite de l'environnement qui nous permet de trouver rapidement des classes, méthodes, attributs, etc.

4 – est la partie du bas de l'environnement qui corresponde aux sorties des divers outils (fenêtre d'exécution du programme, documentation du programme, débogueur).

[36] Eclipse: http://www.eclipse.org/

3.5.3. Outils de développement d'interface

L'interface graphique (en. Graphical User Interface, GUI) réalise la communication entre le système est l'utilisateur. L'objectif d'une interface est d'être ergonomique, efficace, facile à utiliser, adaptée au contexte et de réaliser une bonne communication.

Parmi les outils existants dans le domaine d'implémentation d'interface, nous avons utilisé dans le développement de l'interface du système de consultation de la base de données en langage naturel pour le logiciel EXPESURF : PHP, HTML et JavaScript.

3.5.3.1. HTML

HTML (HyperText Markup Language) est un langage utilisé pour la création des pages Web. Il décrit le format où des éléments sont distribués et vus sur Web, étant un élément fondamental du WWW (Word Wide Web).

HTML a été développé en 1989 par Tim Berners-Lee au CERN. Le but a été de changer d'informations par l'intermède d'Internet entre les physiciens qui utilisent des ordinateurs différents.

HTML est très utilisé grâce à ces caractéristiques :

- facile à utiliser par l'existence des tags. Le tag est une lettre ou un mot décrit entre les « < » et « > » dont les browsers savent l'interpréter tel qu'afficher des images, texte bouton, etc. ;
- l'indépendance de plateforme qui offre la possibilité qu'un document être affiché d'une façon similaire sur des ordinateurs différents ;
- la possibilité hypertexte. Hypertexte signifie que les éléments du document peuvent faire référence vers un autre document.

Le standard officiel HTML est World Wide Web Consortium[37] (W3C). W3C a énoncé jusqu'au maintenant plusieurs variantes de HTML dont: HTML 2.0, HTML 3.0, HTML 3.2, HTML 4.0 et la plus récente HTML 5.0.

3.5.3.2. PHP

PHP[38] (Hypertext Preprocessor) est un langage de scripting réalisé spécial pour le développement des applications Web. Il a été créé en 1994 par le Rasmus Lerdorf.

PHP est un langage de programmation Web, un Open Source utilisé dans le développement des applications server-side[39]. PHP permet de gérer un contenu Web

[37] W3C – HTML : www.w3schools.com/html/default.asp

[38] PHP: http://php.net

[39] Scripting server – side est une technologie Web où les exigences sont accomplir par l'exécution d'un script sur le serveur web . Il est utilisé pour la création des sites web dynamiques qui communique avec les bases de données.

dynamiquement[40]. Le contenu Web est un élément important pour améliorer le trafic d'un site. Par exemple les visiteurs ne reviennent pas sur une page Web qui contient les mêmes informations avec celles déjà vues.

La Figure 26 présente le fonctionnement d'un système qui utilise PHP. Le client envoie les requêtes au serveur Apache. PHP interagit avec les bases de données pour afficher ou modifier les informations et envoie les résultats au serveur qui fournie au client seulement des pages HTML.

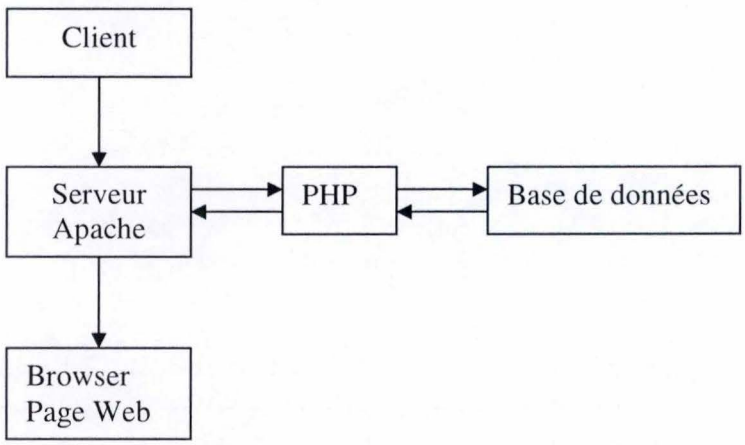
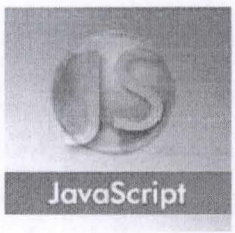


Figure 26 : Fonctionnement PHP

Les principaux avantages d'outil sont qu'il permet l'interaction avec des nombreux systèmes de gestion de base de données relationnels (MySQL, Oracle, Microsoft SQL Server, PostgreSQL), est compatible avec la majorité des systèmes d'exécution (Unix, Linux, Windows, Mac OS X) et interagit avec la majorité des serveurs web.

3.5.3.4. JavaScript



JavaScript est un langage de programmation orienté objet, basé sur les prototypes. Il a été créé par Netscape (Brendan Eric) pour développer des sites interactifs.

Contraire à son nom, JavaScript ne doit pas être confondu avec Java. Ils sont des langages indépendants, ayant en commun la syntaxe qui est proche de C.

Les scripts JavaScript sont introduits dans le code source de HTML et sont interprétés et exécutés par le browser. Ils offrent la possibilité de contrôler le contenu des pages en fonction de date, d'heure, de système d'exploitation ou de browser d'utilisateur. Une autre application est celle-là de construire des sites interactifs qui communique avec les visiteurs ou de valider les données reçues des formulaires.

Le standard officiel JavaScript est World Wide Web Consortium[41] (W3C). Aux dernières années JavaScript est devenu un langage très populaire.

---

[40] Un contenu Web qui se modifie automatiquement.
[41] W3C – JavaScript: www.w3schools.com/js/default.asp

3.5.4. Outils de traitement automatique du langage naturel

Les outils de traitement automatique du langage naturel permettent une amélioration du niveau de traitement et de la compréhension du langage naturel. À l'heure actuelle, ils mettent accent sur le niveau syntaxique. Malheureusement au niveau sémantique et pragmatique ils sont moins performants.

Quel outil pour quoi faire ?

L'Association pour le Traitement Automatique des Langues (ATALA) a mis en place sur leur site internet[42] un répertoire d'outils pour le Traitement automatique des langues. On voit dans l'annexe G que la plupart de ces programmes ont été développés pour réaliser un ou plusieurs traitements linguistiques : l'analyse syntaxique, l'extraction des termes, l'étiquetage, la gestion du lexique, le traitement du corpus.

Parmi l'ensemble des logiciels en ligne ont été retenus par DyALog et Fips.

3.5.4.1. DyALog

DyALog[43] est un environnement pour compiler et utiliser des programmes logiques et des analyseurs syntaxiques tabulaires pour les langues naturelles, créé par Eric de la Clergerie. C'est un projet d'ATALA qui s'occupe avec la compréhension automatique de textes en français. ALPAGE (Analyse Linguistique Profonde A Grande Echelle) a été créée comme Equipe Projet INRIA (EPI) en Juillet 2007 au Centre de Recherche INRIA Paris-Rocquencourt.

DyALog présente l'analyse syntaxique d'une proposition prenant en charge divers formalismes grammaticaux :

- DCG (Definite Clause Grammars);
- BMG (Bound Movement Grammars);
- TAG (Tree Adjoining Grammars);
- TIG (Tree Insertion Grammars);
- RCG (Range Concatenation Grammars

Son site internet présente les principales caractéristiques qui déterminent un traitement plus efficient des parseurs:

- Niveaux différents de tabulation ;
- Stratégies de parsage modifiable en fonction d'utilisateur ;
- Support pour les stratégies de parsage bidirectionnel ;
- L'analyse des listes PROLOG par des automates finies ;
- Filtrage des parseurs lexicaux pour ne charger que des fragments de grammaire ancrée par des mots de la phrase d'entrée ;
- L'étiquetage des constituants pour une meilleure lecture des éléments de sortie.

[42] http://www.atala.org/-Outils-pour-le-TAL- , Annexe G
[43] DyALog : http://dyalog.gforge.inria.fr/

3.5.4.2. Fips

Fips[44] est un outil d'étiquetage créé par LATL (Laboratoire d'Analyse et de Technologie du Langage). LATL est un centre de recherche en linguistique informatique qui a pour vocation le traitement du langage naturel dans les domaines de l'analyse, de la génération et de la traduction automatique.

Selon les présentations du site de LATL, Fips est un projet qui vise à développer un analyseur syntaxique puissant, susceptible d'utilisations pratiques dans le domaine du traitement automatique du langage, et en particulier en traduction assistée par l'ordinateur ou en traitement de la parole.

Fips peut être utilisé online par une interface qui offre la possibilité de choisir la langue (français , l'anglais, l'allemand, l'espagnol et aussi le grec) et le type d'application (Parser, Tagger, Xml). Ces-ci sont présentés dans la Figure 27.

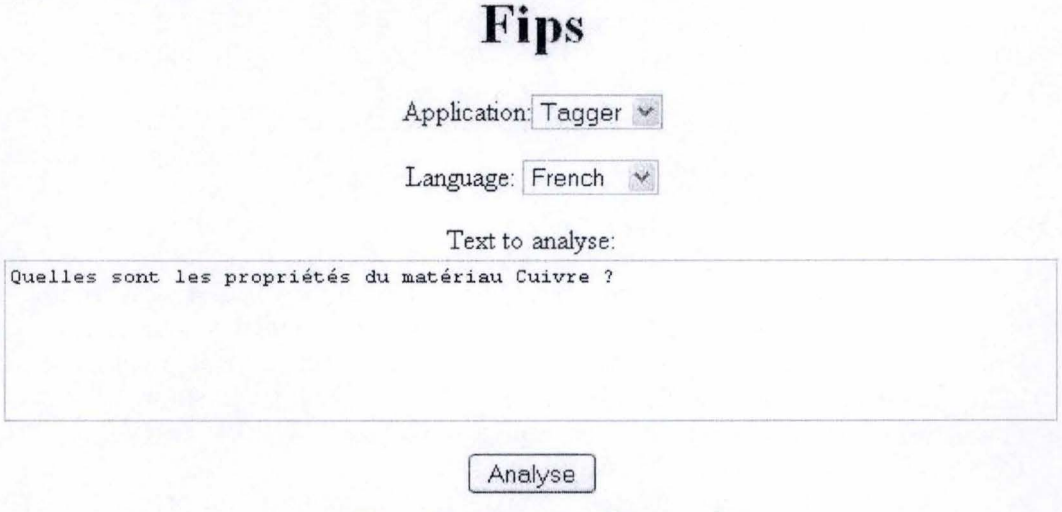


Figure 27 : L'interface Fips

La Figure 27 présente l'analyse fournie par Fips pour notre exemple : « Quelles sont les propriétés du matériau Cuivre ? »

Results:

```
Quelles PRO-INT-PLU-FEM 211049516       1       quel            SUBJ
sont    VER-IND-PRE-3-PLU       211000095       9       être            SUB:quelles     FPO:propriétés
les     DET-DEF-PLU-FEM 211045001       14      le              FPO
propriétés      NOM-COM-PLU-FEM 211014625       18      propriété
du      PRE-CON-de      211045023       29      du
matériau        NOM-COM-SIN-MAS 211020439       32      matériau
Cuivre  NOM-PRO-SIN-ING 0       41      Cuivre
?       PONC-interrogation      0       48      ?
```

Figure 28 : L'analyse syntaxique du Fips

La Figure 28 montre l'analyse syntaxique réalisée par FIPS pour notre question « Quelles sont les propriétés du matériau Cuivre ? ». Pour chaque mot de la question, il identifie le type de mot grammatical (prénom interrogatif, verbe, déterminant, nom commun ou propre, préposition, signe de ponctuation.), le genre et le nombre.

[44] FIPS: http://www.latl.unige.ch/

3.5.5. Le système de question-réponse pour le logiciel EXPESURF et les outils utilisés

Le logiciel EXPESURF utilise une base de données PostgreSQL et une interface réalisée en PHP, HTML et JavaScript. Un système de question-réponse donne la possibilité d'améliorer les performances du logiciel EXPESURF dans le traitement des surfaces. Ce système a été implémenté en utilisant le langage de développement Java. Le lien entre le jar de l'application et l'interface est réalisé d'une manière très facile par l'intermède d'une seule commande PHP.

	Langage
Base de données	PostgreSQL
Interface	PHP, HTML, JavaScript
Système QA	Java

Figure 29 : Outils utilisés en EXPESURF

Pourquoi Java ?

Nous avons utilisé Java comme langage de développement pour sa simplicité, sa possibilité d'utiliser des objets et de traiter des listes. Il permet par conséquent de concevoir un logiciel bien structuré, modulable et maintenable facilement. En plus il améliore les performances des algorithmes pour le traitement de la question. La portabilité, la facilité de la communication avec la base de données et la liaison facile avec le code PHP de l'interface sont des avantages qui nous ont conduits à utiliser Java. Le nombre impressionnant de librairies disponibles pour Java a également orienté notre choix vers ce langage.

Outre le fait que Java soit un logiciel open source, il faut également savoir que Java est utilisé dans le développement des systèmes de question-réponse fermés. Par exemple : « Une bibliothèque d'opérateurs linguistiques pour la consultation de base de données en langue naturelle » -[BOUCHOU, 1999].

3.6. Conclusion

Dans ce chapitre nous avons exploré les éléments techniques de l'élaboration d'un système de question-réponse. En fonction du type du domaine, il y a deux méthodes d'implémentation : *shallow* pour les domaines restreints et *deep* pour les domaines ouverts.

Le fonctionnement des systèmes de question-réponse se réalise dans une séquence des trois étapes : analyse de la question, recherche des documents et extraction des réponses. Après la présentation d'une architecture générale, nous avons décrit les campagnes d'évaluation, ayant comme référence les articles [EL-BEZE, 2006] et [MORICEAU, 2009]. Il y a deux principales manières d'évaluer les systèmes de question-réponse : Moyenne de l'Inverse du Rang, la mesure automatique adoptée par la Moyenne des Rangs Réciproques et le jugement humain

À la fin, nous avons présenté les principaux outils qui peuvent être utilisés dans le développement des systèmes de question-réponse, en mettant en évidence leurs points forts et argumenté notre choix d'outil d'implémentation pour le système d'interrogation en langage naturel pour le logiciel EXPESURF.

Dans le chapitre suivant, nous présentons une étude de cas appliqué au logiciel EXPESURF. Nous réalisons un traitement syntaxique de la phrase interrogative et nous mettons en pratique les étapes présentées dans la partie 3.2. pour un domaine restreint, en appliquant la méthode de type *shallow*.

Chapitre 4

Étude de cas appliqué au logiciel EXPESURF

4.1. Introduction

Dans ce chapitre nous abordons les systèmes de question-réponse de point de vue de leur implémentation dans le logiciel EXPESURF. Au début, nous introduisons notre logiciel, son objectif, sa structure, la base de données et la nécessité d'un système de question-réponse. Puis nous réalisons une étude de la grammaire de langue française, plus précise de la phrase interrogative. Ensuite nous présentons notre implémentation pour la consultation de la base de données en langage naturel pour EXPESURF, les composants, les approches grammaticales, les cas de tests et ses limites. Ce chapitre se conclut par un passage en revue des articles utilisés et de notre contribution.

4.2. EXPESURF

La recherche du mémoire est orientée vers le développement d'un système d'interrogation de la base de données en langage naturel au sein du logiciel EXPESURF. Cette section présente l'architecture générale d'un système expert, l'architecture EXPESURF, la structure de notre base de données et le besoin d'un système de question-réponse.

4.2.1. La structure du logiciel EXPESURF

EXPESURF est un système logiciel expert en surface engineering qui fournie des informations concernant le choix des traitements des surfaces des matériaux à appliquer sur une nouvelle pièce et l'amélioration des propriétés selon les besoins.

Dans cette sous-section nous présentons l'architecture générale d'un système expert et l'architecture du logiciel EXPESURF.

Architecture générale d'un système expert

Un système expert est une application qui prend des décisions ou qui résoudre des problèmes d'un certain domaine ayant à la base des connaissances et des règles analytiques établies par les experts.

Selon [CARSTOIU, 1994] nous présentons la structure d'un système expert en six composants repris sur la Figure 30.

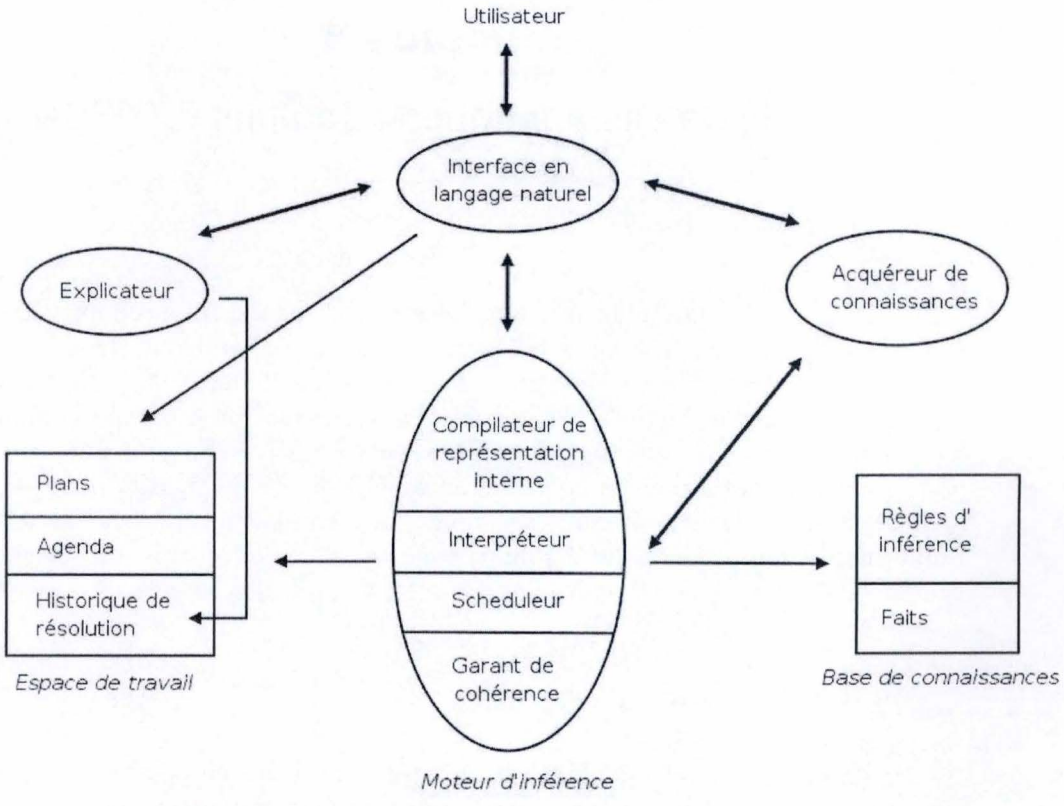


Figure 30 : L'architecture d'un système expert

∞ Base de connaissances
La base de connaissance contient les règles et les faits spécifiques au l'univers du domaine pour lequel on applique le système expert.

∞ Moteur d'inférence
Le moteur d'inférence est le cerveau d'un système expert. Il est un ensemble des procédures avec le but de manipuler la base de connaissances pour effectuer des rationnements sur base du contenu et élaborer le plan pour la résolution du problème.

∞ Interface
Les systèmes experts utilisent une interface en langage naturel pour réaliser une communication conviviale entre utilisateur et système.

∞ Espace du travail
L'espace de travail est une structure de données qui contient le plan des stratégies à appliquer, l'historique de résolution (les valeurs calculées, les objectifs atteints, les décisions prises) et l'agenda (les actions et les règles à appliquer). L'état courant se réfère à ce que le système a déjà fait, ce qu'il va faire et ce qui lui reste à faire.

∞ Module d'acquisition de connaissance
Le module d'acquisition de connaissance utilise les connaissances spécialisées fournies par l'expert humaine.

∞ L'explicateur
L'explicateur permet de tracer le chemin à suivre dans la résolution des problèmes et dans la justification pour les solutions obtenues en soulignant la raison d'erreurs ou d'échec.

Architecture EXPESURF

Dans le cadre du logiciel EXPESURF cette architecture a pris une forme particulière qui est présentée dans la Figure 31.

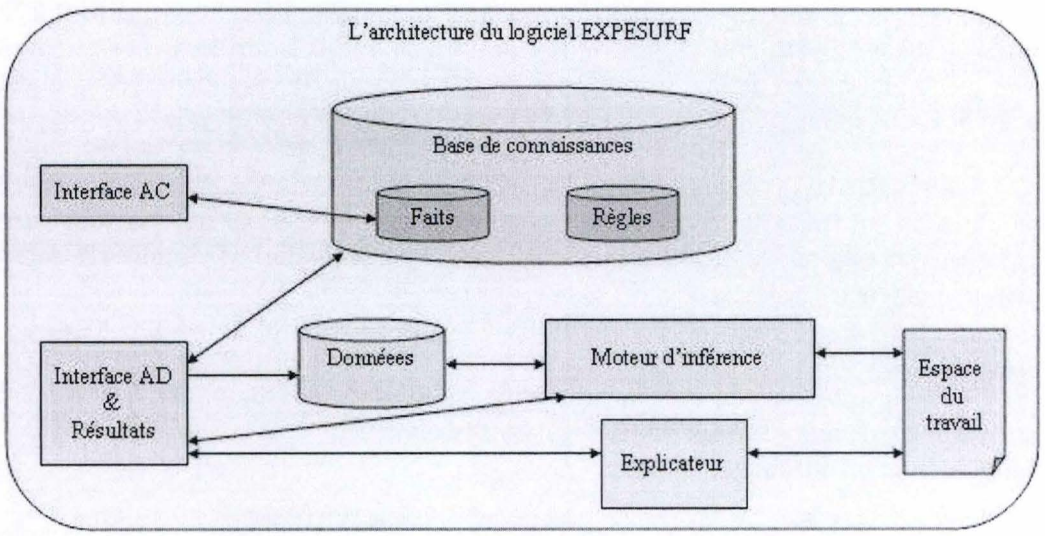


Figure 31 : L'architecture du logiciel EXPESURF

Nous observons que le logiciel EXPESURF respecte la structure d'un système expert. La Figure 31 introduit une particularité du système: le module d'Acquisition des Données. Son but est présenté dans le rapport scientifique du logiciel EXPESURF.

∞ Logiciel Acquisition de Données (AD)

« *Le but du processeur d'acquisition de données est de permettre d'interroger l'utilisateur en vue d'acquérir les informations nécessaires pour effectuer une procédure d'aide au choix de traitements de surface et de construction d'une solution multicouche.* » ([Rapport 3 EXPESURF])

Le processeur d'Acquisition de Connaissances constitue le noyau de notre étude. Le document [Rapport 3 EXPESURF] présente le but du module AC.

∞ Processeur Acquisition de Connaissances (AC)

« Le but du processeur d'acquisition de connaissances est de permettre d'accéder à la banque de données (BD) pour introduire des connaissances dans le système. Il s'agit donc d'une interface pour les utilisateurs qui s'appuie sur la structure de la BD. » ([Rapport 3 EXPESURF])

4.2.2. La structure de la base de données

Le domaine d'ingénierie des surfaces est une sous-discipline de la science de matériau qui recherche la structure des matériaux, ainsi que l'amélioration et l'obtention de nouveaux matériaux avec de propriétés précises et reproductibles ([TADEUSZ, 1999]).

Après l'interaction avec l'environnement, la surface d'un matériau solide peut se dégrader au fil du temps, en générant l'usure, la corrosion, la fatigue et le fluage. Selon l'article [TADEUSZ, 1999], le but de l'ingénierie des surfaces est de réduire cette dégradation, en étudiant les propriétés des surfaces et développant des solutions et applications industrielles pour améliorer les propriétés.

Le matériau est le cœur du domaine d'ingénierie du traitement des surfaces. Il peut être un substrat (la nature d'une pièce qui suit à être traitée) ou une couche (la surface appliquée par le traitement). Les propriétés d'un matériau (dureté, résistance aux frottements, résistance aux chocs, résistance à la corrosion, conductivité thermique) sont améliorées par l'application des procédés qui transforment la structure d'un matériau par l'ajout des couches superficielles et revêtements. Ces procédés sont classifiés en utilisant des critères éliminatoires (éliminent les procédés qui ne respectent pas les exigences) ou Prométhée (classifient les procédés techniquement réalisables).

Souvent au cours d'un traitement de surface de matériau apparaissent des incompatibilités causées par les interactions entre les matériaux. L'ingénierie de surface développe des solutions pour résoudre les incompatibilités entre les matériaux ou entre les procédés.

Toutes ces informations sont comprises dans un schéma de la base de données. Le schéma de base de données est une représentation graphique d'une base de données qui contient un ensemble de concepts, leurs caractéristiques et les relations entre ces-ci. La Figure 32 représente le schéma de la base de données du logiciel EXPESURF.

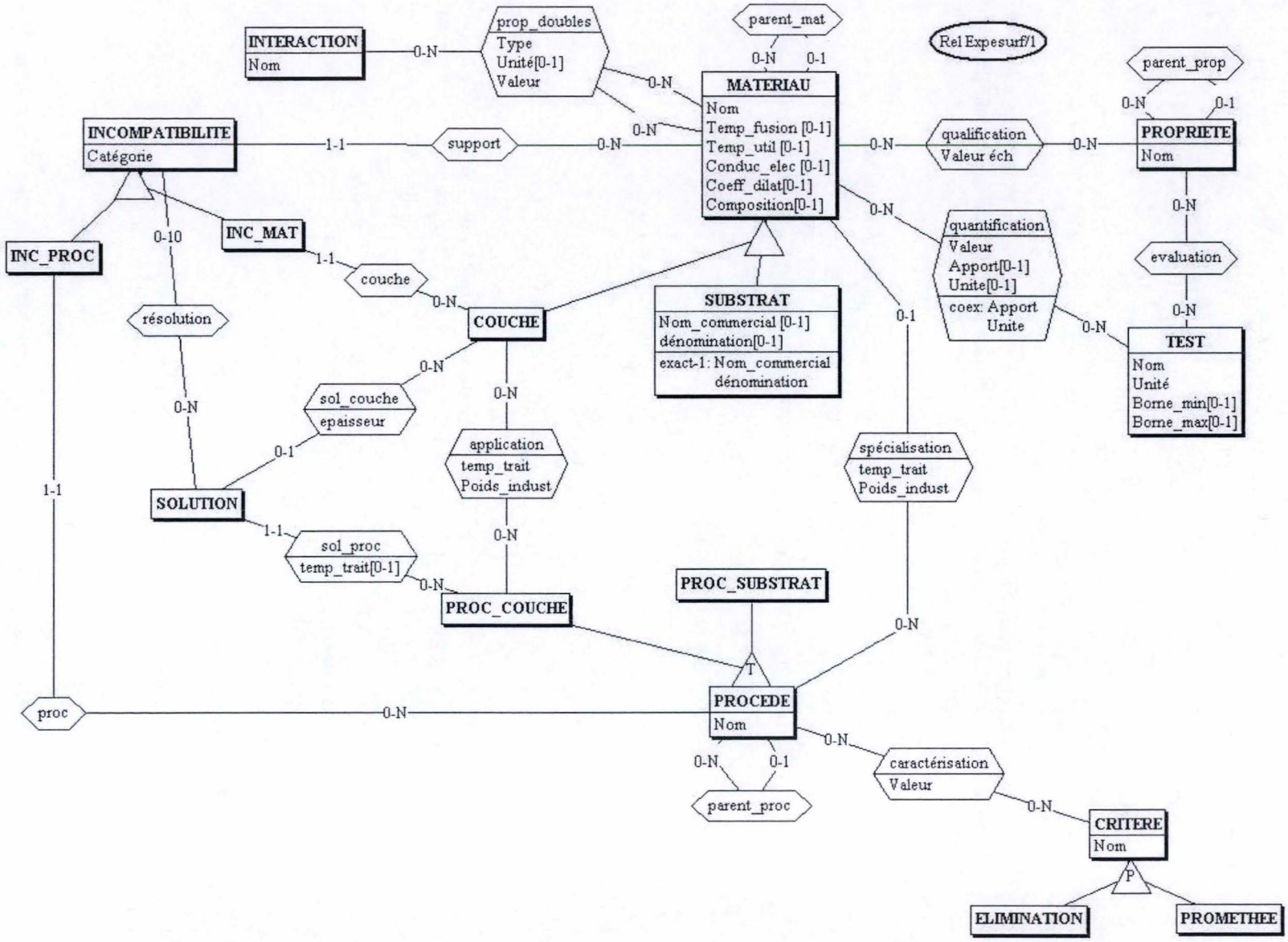


Figure 32 : Le schéma de la base de données EXPESURF

Les principaux concepts illustrés dans le schéma de la base de données du projet EXPESURF sont les matériaux, les procédés, les propriétés, les incompatibilités, les propriétés, les tests et les interactions. En consultant le rapport scientifique, [Rapport 3 EXPESURF], on groupe ces entités dans quatre gestionnaires : gestion des matériaux, gestion des procédés, gestion des propriétés et gestion des incompatibilités.

4.2.2.1. Gestion des matériaux

Le rapport scientifique définit l'*entité matériau* comme étant « *la nature des différents matériaux qui sont utilisés en traitement de surface* ». La même source précise qu'un matériau peut être :

- substrat
  « *La nature d'une pièce destinée à subir un traitement de surface.* »

- couche
  « *La nature d'une couche qui sera appliquée au moyen d'un procédé de traitement de surface.* »

Table 1: Tableau Matériau

MATERIAU
Nom
Temp_fusion [0-1]
Temp_util [0-1]
Conduc_elec [0-1]
Coeff_dilat [0-1]
Composition [0-1]

La Table 1 présente les caractéristiques d'un matériau : nom, composition, températures de fusion, température d'utilisation et propriétés physiques spécifiques (conductivité électrique, coefficient de dilatation). On observe que seulement le nom est obligatoire, les autres attributs étant optionnels.

En ce qui concerne les associations avec les autres entités, l'entité matériau interagit avec interaction, propriété, test, procédé, incompatibilité et solution. Ces associations sont expliquées plus bas.

4.2.2.2. Gestion des propriétés

La gestion des propriétés est réalisée par l'interaction des trois entités de la base de données: entité propriété, entité test et entité interaction.

Entité propriété

L'entité propriété est associée aux propriétés spécifiques pour les matériaux ou aux propriétés apportées par un nouveau traitement.

Table 2: Table Propriété

PROPRIETE
Nom

La Table 2 présente l'entité propriété de la base de données. On observe qu'elle possède un seul attribut : le nom de la propriété.

Concernant cette entité, elle interagit avec l'entité matériau et l'entité test par les associations suivantes:

∞ Association propriété-matériau

La liaison entre le matériau et la propriété est réalisée par une table *qualification* qui est définie par une valeur en échelle (OUI/NON). *« Les matériaux NON, associés à une valeur nulle, ne sont pas présents dans cette table. Les matériaux OUI sont évalués dans cette échelle par rapport à des matériaux de référence qui constituent les échelons. »* [Rapport 3 EXPESURF]

Table 3: Table Qualification

QUALIFICATION
Valeur éch.
Ref_mat
Ref_prop

La Table 3 présente la table qualification de la base de données. Elle contient aussi les références des tables matériau et propriété.

∞ Association propriété-test

Elle est présentée ci-dessous.

Entité test

L'entité test représente le cas d'évaluation d'une propriété.

Table 4: Table Test

TEST
Nom
Unité
Borne_min [0-1]
Borne_max [0-1]

La Table 4 présente l'entité test de la base de données. Elle est caractérisée par les attributs : nom, unité, bornes minimales et maximales liées à ce test. Pour l'entité test, on identifie deux types des associations présentées ci-dessous :

∞ Association test-matériau

Cette association est représentée par la table *quantification* (présenté dans la Table 5). Elle est caractérisée par une valeur, un apport et une unité.

« La valeur chiffrée est en relation directe avec l'unité associée à un test. Ce lien fait également apparaître une notion d'épaisseur de couche pour laquelle la propriété peut être prise en compte. L'unité utilisée pour exprimer l'épaisseur de couche est également mentionnée. » [Rapport 3 EXPESURF]

Table 5: Table Quantification

QUANTIFICATION
Valeur
Apport [0-1]
Unité [0-1]
Ref_mat
Ref_test

∞ Association test-propriété
Cette association est représentée par la table *évaluation* et définit par quels tests une propriété peut être évaluée.

Table 6: Table Evaluation

EVALUATION
Ref_test
Ref_prop

La Table 6 présente la table évaluation de la base de données. Cette table n'a pas des attributs spécifiques, mais contient les références des tables avec lesquelles interagit.

Entité interaction

L'entité interaction représente les propriétés d'une interaction entre deux matériaux.

Table 7: Table Interaction

INTERACTION
Nom

La Table 7 présente la table interaction de la base de données. Elle est caractérisée par l'attribut nom et interagit avec l'entité matériau. La liaison est réalisée entre une interaction et deux matériaux définissant le type de la propriété double, son unité et sa valeur (décrites dans la Table 8).

Table 8: Table Prop_doubles

PROP_DOUBLES
Type
Unité [0-1]
Valeur

4.2.2.3. Gestion des procédés

La gestion des procédés est réalisée par l'interaction des deux entités de la base de données: entité procédé et entité critère.

Entité procédé

L'entité procédé représente le moyen par lequel on peut transformer la structure d'un matériau ou appliquer une nouvelle couche. Le [Rapport 3 EXPESURF] identifie deux types des procédés:

∞ Substrat
« Procédé destiné à être appliqué au substrat tant qu'aucune couche n'a été déposée au préalable. »

∞ Couche
« Procédé destiné à être utilisé pour réaliser une couche ou pour traiter une couche préalablement réalisée. »

Table 9: Table Procédé

PROCEDE
Nom

La Table 9 présente l'a table procédé de la base de données. Elle est caractérisée par un seul attribut : le nom, mais elle interagit avec trois entités : critère, matériau et incompatibilité. Donc on distingue trois types d'associations :

∞ Association procédé-critère
Cette association est représentée par la table caractérisation.

Table 10: Table Caractérisation

CARACTERISATION
Valeur

Comme vue dans la Table 10 la caractérisation des procédés par les critères est exprimée par une valeur.

∞ Association procédé-matériau
Cette association est représentée par deux tables spécialisation et application.

Table 11: Table Spécialisation

SPECIALISATION
Temp_trait
Poid_indust

La Table 11 représente la table spécialisation de la base de données. Elle réalise le lien entre l'entité matériau et l'entité procédé.

Table 12: Table Application

APPLICATION
Temp_trait
Poid_indust

La Table 12 représente la table application de la base de données. Elle fait le lien entre la couche du matériau et l'entité procédé. Les caractéristiques de ce traitement sont la température du traitement et le poids industriel.

∞ Association procédé-incompatibilité
Cette association est décrite plus bas.

Entité critère

L'entité critère représente les caractéristiques des procédés qui permets les classifiés.

Table 13: Table Critère

CRITERE
Nom

La Table 13 présente la table critère de la base de données. Elle est caractérisée par un seul attribut : le nom et interagit avec la table procede (l'association décrite plus haut).

Le rapport scientifique, [Rapport 3 EXPESURF], identifie deux types des critères :

- Eliminatoire
  « *Critère destiné à éliminer les procédés présentant des contraintes de mise en œuvre incompatibles avec les exigences de l'utilisateur.* »
- Prométhée
  « *Critère destine à effectuer un classement entre les différents procédés techniquement réalisables.* »

4.2.2.4. Gestion des incompatibilités

La gestion des procédés est réalisée par l'interaction des deux entités de la base de données: entité incompatibilité et entité solution.

Entité incompatibilité

L'entité incompatibilité représente l'impossibilité qu'un matériau être utilisable dans un certain cas. [Rapport 3 EXPESURF] identifie quatre catégories d'incompatibilités (Matériau – Couche, Couche – Procédé, Procédé – Matériau, Matériau – Couche – Procédé) qui sont groupés en :

- Incompatibilités matérielles
  « *Ce groupe représente une incompatibilité entre deux matériaux dont un au moins appartient au groupe couche de l'entité matériau.* »
- Incompatibilité de procédé
  « *Ce groupe représente une incompatibilité entre un procédé et un matériau pouvant appartenir au groupe couche ou substrat de l'entité matériau.* »

Table 14: Table Incompatibilité

INCOMPATIBILITE
Catégorie

On observe dans la Table 14 que l'entité incompatibilité de la base de données est caractérisé par l'attribut catégorie. Elle interagit avec les tables matériau, solution, procédé, les associations décrites au-dessous.

- Association incompatibilité-matériau
  Cette association est représentée par deux liens : l'une avec l'entité matériau (le support pour lequel on cherche l'incompatibilité avec le traitement), l'autre avec la couche de l'entité matériau pour permettre les identifications d'incompatibilité du matériau.
  Celle-ci est réalisée par les tables *support*, respectivement *couche*.

- Association incompatibilité-procédé
  Cette association est réalisée entre une entité procédé et une entité incompatibilité et a comme but l'identification d'une incompatibilité de type procédé.
  Dans la base de données elle est représentée par la table *proc*.

- Association incompatibilité-solution
  Cette association est décrite plus bas.

Entité solution

L'entité solution représente une façon de résoudre les incompatibilités entre les matériaux, procédé ou couche.

Table 15: Table Solution

SOLUTION
Type
Epaisseur
Temp_trait

La Table 15 présente l'entité incompatibilité de la base de données. Elle est caractérisée par un type, l'épaisseur et la température de traitement. Ses interactions avec les tables incompatibilité, matériau et procédé sont présentés dans les associations suivantes:

- Association solution-incompatibilité
  Cette association entre l'entité solution et l'entité incompatibilité est représenté par l'entité *résolution* qui signifie l'existence d'une réponse à l'incompatibilité rencontrée. On observe dans la Table 16 que la table résolution ne contient pas des attributs propres, mais seulement des références vers les tables avec quelles interagit.

Table 16: Table Resolution

RESOLUTION
Ref_inc
Ref_sol

- Association solution-procédé
  Cette association est représentée par la table *sol_proc*. Sa structure, présentée dans la base de données, est décrite dans la Table 17.

Table 17: Table Sol_Proc

SOL_PROC
Temp_trait [0-1]

La Table 17 est caractérisée par la température de traitement qui doit être utilisé quand on applique le procédé pour résoudre l'incompatibilité.

∞ Association solution-matériau
Cette association représente que la solution de l'interaction est une couche avec une certaine épaisseur. Celle-ci est présentée dans le schéma de la base de données par la table *sol_couche* (présenté dans Table 18).

Table 18: Table Sol_Couche

SOL_COUCHE
Epaisseur

Jusqu'ici on a introduit la structure de la base de données et on a décrit la signification des tables. Ensuite on présente pourquoi notre système, EXPESURF, a besoin d'une interface d'interrogation de la base de données en langage naturel et comment on l'implémente.

4.2.3. Pourquoi un système de question-réponse ?

Dans le contexte du traitement de surface, les exigences fonctionnelles augmentent et les contraintes de l'environnement sont difficiles d'être respectées durant l'élaboration, en cours d'utilisation et de vie des pièces. En général, un seul traitement n'est pas suffisant, donc se réalise plusieurs couches successives par multitraitements. Comment savoir toutes les interactions et les propriétés de la couche antérieure sans être un spécialiste dans les bases de données du logiciel?

L'interface est le moyen par l'intermédiaire duquel l'utilisateur accède à la base de données pour connaître ces informations. Quelle interface peut-être plus conviviale qu'une interface en langage naturel ? Le langage naturel est une modalité de communication homme-machine qui présente des avantages majeurs : ne nécessite pas une formation ou une familiarisation avec le langage. Il est un moyen direct d'accès à l'information, indépendant de la structure et la codification de celle-ci.

Donc une interface en langage naturel résout ces problèmes dans le cadre du logiciel EXPESURF, permettant le traitement d'une grande quantité d'informations dans un délai plus court et l'ouverture générale vers les clients qui n'ont pas une formation spécifique.

Dans les sections suivantes, nous présentons l'implémentation de notre système de question – réponse pour le logiciel EXPESURF. La section 4.3. étudie la grammaire de la phrase interrogative, en mettant accent sur la syntaxe de la question. La section 4.4. décrit notre implémentation d'un système d'interrogation de base de données en langage naturel pour le logiciel EXPESURF.

4.3. Grammaire : la phrase interrogative

« *La phrase interrogative est une des modalités d'énonciation qui correspond à une attitude énonciative non thétique (le locuteur demande une information ou une validation) et à un acte de langage (celui de la question)* ».[45] La langue française est une langue romaine, riche de points de vue de la structure de phrase interrogative.

Dans le chapitre 3 nous avons vu que le module d'analyse de la question est une étape essentielle dans la réalisation des systèmes question-réponse. À la base de notre recherche se trouve un traitement présyntaxique de la question en langue française. Elle permet de récupérer les informations essentielles pour identifier les termes importants, le type de la question et le type de la réponse. Pour obtenir des réponses correctes, il faut réaliser une analyse plus détaillée de la question.

Dans cette section, nous abordons la grammaire de la phrase interrogative. Nous parlons de type, de la structure, de déclencheurs et des restrictions d'une question en langue française.

4.3.1. Type de la phrase interrogative

Conforme l'article « Phrases interrogatives directe (type interrogatif) et indirecte »[46] nous distinguons deux types de phrases interrogatives :

Phrase interrogative directe

À sa place, la phrase interrogative directe est de deux types :

∞ Phrase avec interrogation totale (ou globale)

Phrase avec interrogation totale (ou globale) est la phrase où l'interrogation porte sur tout l'ensemble de la phrase. Elle est une demande de validation, donc la réponse peut être « oui » ou « non».
La phrase interrogative totale se termine par un point d'interrogation et est caractérisée par une intonation ascendante.

Exemple tiré du domaine EXPESURF :
Est-ce que le matériau Cuivre possède la propriété « brillant » ?

∞ Phrase avec interrogation partielle

Phrase avec interrogation partielle est la phrase où l'interrogation porte sur un élément de la phrase qui est représenté par mot interrogatif. Elle est une demande d'information, donc la réponse n'est pas une affirmation positive ou négative, mais un ou plusieurs éléments précis.

---

[45] Grammaire – L'interrogation : http://www.etudes-litteraires.com
[46] Rubrique grammaticale du site : http://www.ccdmd.qc.ca/fr

La phrase interrogative partielle se termine par un point d'interrogation et est caractérisée par une intonation descendante, mettant accent sur le mot interrogatif.

Exemple tiré du domaine EXPESURF:

Quels sont les matériaux qui ont la propriété « brillant » ?

Phrase interrogative indirecte

La phrase interrogative indirecte est contenue dans l'intérieur de la phrase. Elle ne se termine pas par un point d'interrogation, mais elle est marquée par la présence des verbes : se demander, chercher, ignorer, savoir, etc.

Exemple tiré du domaine EXPESURF:

Je me demande si le matériau Cuivre possède la propriété « brillant » .

4.3.2. Structure de la phrase interrogative directe

Dans la réalisation de notre application d'interrogation en langage naturel de la base de données pour le logiciel EXPESURF, nous utilisons une analyse approfondie de l'ordre des mots dans la phrase interrogative directe.

En langue française une phrase interrogative directe est caractérisée par :

- un mot interrogative ;
- une inversion ;
- le point d'interrogation.

En ce qui concerne la syntaxe de la langue française, l'ordre des mots dans une phrase interrogative est inversé. L'inversion est la méthode considérée la plus formelle pour la formation d'une question. Elle consiste dans l'inversion de la place du verbe avec celui du pronom sujet et par l'ajout d'une ligne entre ces deux.

On identifie les suivantes configurations syntaxiques dans une question, en fonction de leur type:

∞ Phrase interrogative totale

Conforme l'article « Grammaire – l'Interrogation » (45) dans une phrase interrogative totale il existe plusieurs configurations syntaxiques :

- interrogation sans inversion : l'interrogation est marquée par une intonation montante ;
- interrogation avec inversion du sujet :
  - inversion simple : Verbe + Sujet ;
  - inversion complexe : Sujet + Verbe – Pronom ;
- interrogation avec la construction est-ce que :

  est-ce que + Sujet + Verbe

∞ Phrase interrogative partielle

Mot Interrogatif + Verbe + Sujet

∞ Phrase interro-négative

Ne + Verbe + Sujet + Pas

Nous observons que pour construire une phrase interrogative partielle il faut utiliser tout d'abord un déclencheur interrogatif devant le verbe. Le sujet indique le type de la réponse et sur quoi porte la question. Donc le mot interrogatif remplace la réponse de la proposition déclarative. L'analyse présyntaxique de notre système de question-réponse se base sur ce rationnement.

4.3.3. Mots interrogatifs de la proposition interrogative

Les phrases interrogatives sont introduites par des « mots interrogatifs » qui s'appellent aussi outils interrogatifs. Leur nature peut être pronom interrogatif, adjectif interrogatif ou adverbe interrogatif.

∞ Pronom interrogatif

Le pronom interrogatif est un outil interrogatif qui sert à introduire une question. La Figure 33 présente les types et la fonction des pronoms interrogatifs : *qui, que, quoi, lequel.*

Pronom	Fonction
qui	Sujet, attribut, complément.
que	C.O.D., attribut, sujet.
quoi	Sujet, complément
préposition + qui	C.O.I.
préposition + quoi	C.O.I.
lequel	Toutes les fonctions

Figure 33 : Les pronoms interrogatifs

Le pronom interrogatif « lequel » indique un choix entre plusieurs éléments et s'accorde avec le nom qu'il remplace. Donc il présente des formes qui varie en genre et nombre : laquelle, lesquels, lesquelles. Il possède aussi des formes composées pour interroger sur l'identité des objets retirés d'un ensemble.

La Figure 34 présente les formes composée des pronoms interrogatifs.

Pronom	à +	de +
quel	auquel	duquel
quelle	à quelle	de quelle
quels	auxquels	desquels
quelles	auxquelles	desquelles

Figure 34 : Les formes composées du pronom interrogatif

∞ Adverbe interrogatif

Les adverbes interrogatifs peuvent être de lieu, de temps, de quantité, de manière ou de cause. On les présente dans la Figure 35.

Adverbe	Type
où	lieu
quand	temps
combien	quantité
comment	manière
pourquoi	cause

Figure 35 : Les adverbes interrogatifs

∞ Adjectif interrogatif

Les adjectifs interrogatifs sert à interroger au sujet d'élément qu'il le détermine. La Figure 36 : Les adjectifs interrogatifs présente les types et la fonction des adjectifs interrogatifs : *combien de, quel, quelle, quels, quelles.*

Adjectif	Fonction
quel, quelle, quels, quelles	Sujet, attribut
combien de	Attribut

Figure 36 : Les adjectifs interrogatifs

4.3.4. Restrictions de la question

On considère comme l'entrée de notre système de question-réponse seulement des phrases interrogatives partielles.

Pour que notre parseur réussir à identifier les mots représentatifs de la question, on introduit quelques restrictions :

- les mots sont délimités par un espace ;
- l'existence d'un mot interrogatif ;
- l'existence de l'inversion Prédicat + Sujet ;
- le verbe au mode indicatif présent.

4.4. Système de question-réponse pour le logiciel EXPESURF

4.4.1. Présentation générale

Cette section présente notre implémentation d'un système d'interrogation de base de données en langage naturel pour le logiciel EXPESURF. Le but de notre application est de traduire une question en langage naturel vers une requête SQL et d'offrir des réponses correctes et précises aux questions. Pour atteindre ces objectifs, nous avons utilisé un dictionnaire de mots-clés et une bibliothèque d'opérateurs (prédicats) spécifiques au domaine d'application du traitement de la surface des matériaux.

Notre approche est une alternative au traitement sémantique et pragmatique présenté dans le chapitre 2. Dans le cadre du logiciel EXPESURF le problème est moins compliqué : le monde de traitement de la surface des matériaux est un domaine restreint, très précis et peu variable. Notre travail est particulier, étant limité en rapport avec le langage naturel complexe. Faisant référence à la section d'analyse de la question présentée dans le chapitre 3 (Figure 9), le système de question-réponse du logiciel EXPESURF est appliqué seulement pour les questions de type liste, requête évaluative, requête comparative et procédures. Dans ce contexte, le type de réponse n'est pas très diversifié. Nous n'avons pas de personnes, d'organisations, de dates ou de lieux, mais seulement de matériaux, propriétés, procédées, critères et tests.

Dans ces hypothèses, il n'y a pas besoin d'une analyse sémantique ou pragmatique, car une analyse approfondie de la syntaxe de la question est suffisante. Parmi les méthodes utilisées dans le développement des systèmes de question-réponse (présentés dans le chapitre 3) nous mettons en pratique la méthode *shallow*, une méthode qui a succès pour les questions courtes et pour les domaines fermés. Les informations morpho-lexicales sont identifiées en utilisant un dictionnaire comme base de connaissances et la requête est construite par l'utilisation d'une bibliothèque d'opérateurs. Nous utilisons donc dans le développement de notre application un système de mots clés, combiné aux connaissances fournies par les opérateurs.

Guide de lecture

La suite de cette section présente en 8 sous-sections le développement de notre application : un système de consultation en langage naturel de la base de données du logiciel EXPESURF. Les deux premières décrivent la structure de notre système, en mettant en évidence les outils créés : un système de question-réponse responsable du traitement de la question et de la construction de la requête, et une interface en langage naturel qui permet de soumettre aisément la question et d'afficher la réponse. Les sous-sections 4.4.4. et 4.4.5. décrivent le dictionnaire des mots clés et les opérateurs, des éléments essentiels pour notre application.

Ensuite, nous présentons un cas de test pour une question simple : « *Quel est le nom du matériau qui possède la propriété Brillant ?* », en parcourant les étapes intermédiaires de l'exécution de notre application. La sous-section 4.4.7. décrit les

limites de notre application. Notre système a été créé de manière à permettre une intégration facile par des autres logiciels. Nous présentons dans la sous-section 4.4.8. la portabilité du système et la structure des fichiers XML, responsables de la population du lexique. La section se termine ensuite par la présentation de la documentation du système.

4.4.2. La structure du système

La structure de notre application d'interrogation en langage naturel de la base de données se compose de deux outils importants : une interface en langage naturel et un système de question-réponse. L'interface réalise la communication avec l'utilisateur d'une manière conviviale grâce au langage naturel, tandis que le système de question-réponse a le but de réaliser le traitement de la question.

La recherche de la bonne réponse de la question est réalisée par l'exécution d'une séquence des étapes présentées dans la Figure 37.

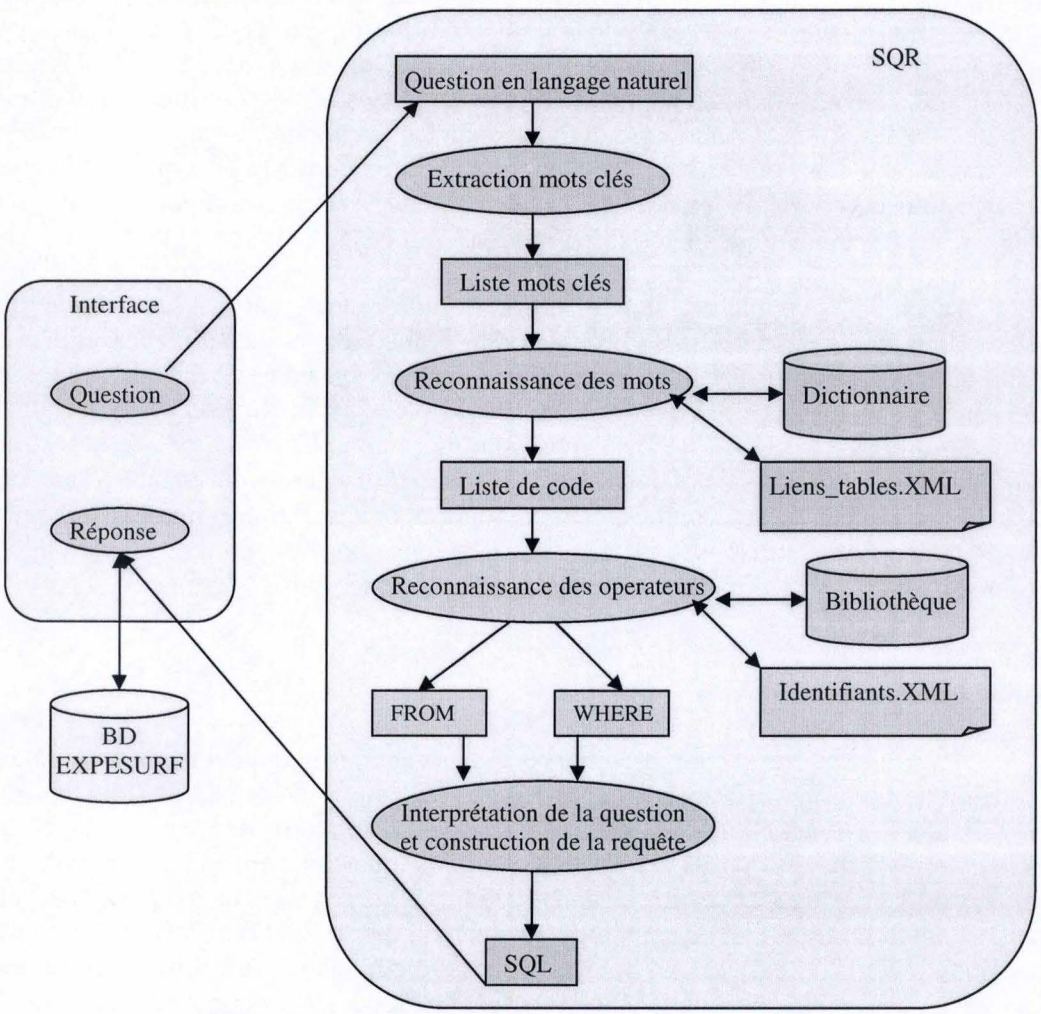


Figure 37: Structure du système de question-réponse EXPESURF

Au début, l'utilisateur du système introduit une question, utilisant l'interface en langage naturel. Cette question est transférée au système de question-réponse qui réalise son traitement. En parcourant des étapes mentionnées au chapitre 3, étapes de l'analyse de la syntaxe de la question, de la reconnaissance des mots clés, de recherche des opérateurs linguistiques, nous obtenons une construction de la requête SQL. Celle-ci est transférée vers l'interface, qui réalise la consultation de la base de données et affiche la réponse qui est visualisée par l'utilisateur.

Les outils et la séquence des étapes présentées dans la Figure 37 sont décrites en détail dans la sous-section suivante.

4.4.3. Les outils créés pour la consultation de base de données en langage naturel

Plusieurs outils ont été créés dans le processus d'acquisition des connaissances du logiciel EXPESURF pour consulter la base de données en langage naturel. Ces-ci sont le système de question-réponse proprement dit qui réalise le traitement de la question et l'interface qui permet d'introduire les questions et d'afficher les réponses. Ils sont décrits en détail ci-dessous.

4.4.3.1. Système de question-réponse

Dans cette sous-section nous décrivons notre système de question-réponse du point de vue de l'implémentation. Ce programme a été écrit en Java et consiste à effectuer un traitement automatique d'une question en langage naturel, en la transformant dans une requête SQL d'interrogation de la base de données EXPESURF. Notre exposé est illustré pas le traitement de la question « *Quel est le nom du matériau qui possède la propriété Brillant ?* »

Le traitement effectué par notre système se réalise suivant la séquence d'étapes présentées dans la Figure 37. Du point de vue du domaine d'analyse, celles-ci sont partagées en deux types: éléments de traitement linguistiques et éléments de traitement informatiques.

Outils de traitement linguistique

Pour le traitement linguistique de la question, le système utilise trois outils linguistiques :

- **dictionnaire de mots-clés**

  Le dictionnaire est construit en analysant le schéma de la base de données, un dictionnaire des synonymes et la liste des opérateurs. Il contient les noms des tables et des attributs, leurs synonymes, les opérateurs et des mots appartenant à leur famille lexicale. Le dictionnaire est utilisé dans la phase d'analyse de la proposition interrogative, ayant pour but de réaliser une reconnaissance des mots clés de la question.

Le dictionnaire associe à chaque entrée un code dont la structure permet d'identifier l'élément (table, attribut ou instance) de la base de données EXPESURF qui lui correspond.

Le dictionnaire est présenté en détail au point 4.4.4.

∞ **bibliothèque des opérateurs**

Nous avons construit la bibliothèque d'opérateurs par la consultation du schéma de la base de données du logiciel EXPESURF et par une analyse des opérateurs linguistiques qui précise la demande dans une proposition interrogative. Le rôle d'un opérateur est de connecter plusieurs tables entre elles, en gardant la logique de la BD. À chaque association entre deux entités de la base de données (les associations étudiées dans la sous-section 4.2.2.) nous créons un opérateur. En plus la bibliothèque contient des opérateurs linguistiques, les mots clés qui fournissent des informations concernant le type de réponse de la question : minimum, maximum, supérieur, inférieur, count, et, etc. La bibliothèque des opérateurs associe à chaque entrée un prédicat.

Les opérateurs sont présentés en détail au point 4.4.5.

∞ **une interprétation de la question**

A un autre niveau, l'interprétation de la question est une étape essentielle dans le développement des systèmes de question-réponse. Elle permet de récupérer les informations pertinentes nécessaires à l'identification de la réponse correcte. Nous réalisons une analyse syntaxique de la phrase interrogative sur base des codes identifiés, en tenant compte de façon d'arrangement des mots dans une question.

Notre application est orientée vers l'utilisation des phrases interrogatives partielles. Dans la section 4.3.2. nous identifions leur structure :

Mot Interrogatif + Verbe + Sujet

Cette structure fournit une information importante : la place du type de réponse (le sujet) en fonction du déclencheur de la question (mot interrogatif).

Le résultat de cette étape est l'identification du type de réponse et la création de la partie *SELECT* d'une requête SQL.

Étapes de traitement

La gestion de la question s'article en quatre étapes importantes :

∞ **l'extraction de mots-clés**

Le premier pas du traitement d'une phrase interrogative est la segmentation. C'est une étape essentielle qui traduit le texte dans une séquence des unités lexicales (mots). Parmi les types de séparateurs existants (présentés dans la Figure 4, chapitre 2), dans notre cas nous utilisons l'espace blanc (« espace »).

La segmentation renvoie une liste avec tous les mots de la question. Nous faisons abstraction des prépositions (de, à, avec, pour, etc.) ou des déterminants (le, la, les), en retenant seulement les mots pertinents. Cette étape envoie vers la suivante une liste de mots clés.

Dans l'exemple de la question:

> Quel est le nom du matériau qui a la propriétés Brillant ?

nous obtenons la liste suivante :

> quel, est, le, nom, du, matériau, qui, a, la, propriété, Brillant, ?

∞ **la reconnaissance des mots du dictionnaire**

Nous réalisons un scannage de la liste avec les mots clés obtenus par l'étape antérieure et parmi ceux-ci nous ignorons les mots qui ne sont pas importants : les prépositions, les déterminants, les signes de ponctuation, etc. Chaque mot qui a été considéré comme pertinent est cherché dans la base de données *dictionnaire* (présente en détail au point 4.4.4.). L'information est associée à une entité de la base de données représentée par un code dans un format spécial : « O . Operateur », « T . *Table* », « *A . Table . Attribut* » ou « *I . Table . Attribut . Instance* ».

La reconnaissance des instances est un cas spécial, à cause du fait qu'elles ne sont pas stockées dans la table Dictionnaire. Pour chaque mot considéré pertinent qui reste inconnu après l'association avec le dictionnaire, nous faisons une recherche dans les tables antérieures des codes de type « T . *Table* » et « *A . Table . Attribut* ». Par exemple pour la construction « *matériau Cuivre* », Cuivre va être cherché dans la table matériau. Nous obtenons les codes : T.materiau, I.materiau.nom.Cuivre.

La question peut être moins précise : « *Quelles sont les propriétés du Cuivre ?*». Dans ce cas il n'est pas précisé le type de l'instance « Cuivre ». Pour résoudre ces problèmes, nous avons créé un fichier « liens_tables.XML » qui définit pour chaque table de la base de données les tables avec lesquelles elle interagit de manière logique. Pour notre exemple l'entité *propriété* interagit avec *matériau* et *test*. Après la recherche nous obtenons comme réponse I.materiau.nom.Cuivre.

Cette étape a comme résultat une liste des codes, chaque code étant la définition d'un mot reconnu.

Prenant l'exemple de notre question « *Quel est le nom du matériau qui possède la propriété Brillant ?* » le système de reconnaissance des mots-clés va trouver les valeurs suivantes :

> quel –> *O . select*
> nom –> *A . materiau . nom*
> matériau –> *T . materiau*
> propriétés –> *T . propriete*
> *Brillant* –> *I . propriete. nom . Brillant*

∞ **la reconnaissance d'opérateurs**

Avec la liste des codes obtenue par l'étape antérieure nous sélectionnons les opérateurs spécifiques pour la question. Nous sélectionnons de la bibliothèque *Operateur*, tous les opérateurs qui ont tous les arguments trouvés parmi les tables des mots-clés. Chaque opérateur définit ensuite une clause FROM et une clause WHERE (souvent optionnelle) de la requête SQL.

Dans notre exemple (« *Quel est le nom du matériau qui possède la propriété Brillant ?* ») l'application sélectionne les opérateurs :

```
QUALIFICATION ( materiau, propriete )
PROPRIETE ( I . propriete . nom . Brillant)
```

Pour la définition de la clause WHERE nous utilisons le fichier XML « Identifiants.XML » qui contient la clé primaire (l'identifiant) et la clé étrangère (la référence) pour chaque table de la base de données EXPESURF. Pour chaque opérateur, le système définit automatiquement de manière générale les conditions suivantes :

```
NOM_OPERATEUR . ref_table1 = NOM_TABLE1 . id_table1
&& NOM_OPERATEUR . ref_table2 = NOM_TABLE2 . id_table2
```

Par exemple pour l'opérateur *qualification (matériau, propriété),* nous avons la clause suivante :

```
WHERE
        QUALIFICATION . ref_mat = MATERIAU . id_mat
        && QUALIFICATION . ref_prop = PROPRIETE . id_prop
```

La clause FROM, décrite par un opérateur, spécifie les tables source pour le SELECT. Le système la fournit de manière automatique, en contenant toutes les tables utilisées dans la clause WHERE de l'opérateur.

```
NOM_OPERATEUR, NOM_TABLE1, NOM_TABLE2
```

Par exemple pour l'opérateur *qualification (matériau, propriété),* la clause FROM sera :

```
FROM
        QUALIFICATION, MATERIAU, PROPRIETE
```

Cette étape renvoie deux listes : l'une qui contient les tables qui vont servir pour la création de la clause *from* de la requête et l'autre avec les conditions pour la clause *where*.

Prenant l'exemple de notre question « *Quel est le nom du matériau qui possède la propriété Brillant ?* » nous obtenons :

QUALIFICATION (materiau , propriete)

FROM MATERIAU, PROPRIETE, QUALIFICATION
WHERE QUALIFICATION . ref_mat = MATERIAU . id_mat
AND QUALIFICATION . ref_prop = PROPRIETE . id_prop

PROPRIETE (I . propriete . nom . Brillant)

FROM PROPRIETE
WHERE PROPRIETE. nom = 'Brillant'

∞ **la construction de la requête**

Une requête SQL commence avec un verbe qui décrit ce qu'il fait l'instruction. Par exemple SELECT, INSERT, DELETE, CREATE. Notre application réalise seulement d'interrogations de la base de données, donc nous utilisons le type SELECT. Ensuite la requête contient des clauses qui spécifient quelles sont les données exploitées ou fournissent des informations concernant l'instruction. Certaines clauses comme WHERE, GROUP BY, ORDER BY et HAVING sont optionnelles, tandis que FROM est obligatoire. Seule la clause WHERE est prise en considération dans ce travail.

L'étape de la construction de la requête consiste dans la création d'une instruction SQL d'interrogation de la base de données. Elle utilise les listes *from* et *where* créées dans les étapes antérieures et réalise une identification de la partie *select*. Le mot-clé de la question qui est de type « O . select » représente le mot interrogatif ou le déclencheur de la question. Il donne une information importante pour la définition de la partie *select* de la requête: le mot-clé qui le suit représente le type de réponse.

Après la concaténation de ces trois listes (select, from, where) le système obtient la requête SQL. Elle est envoyée vers l'interface de l'application, qui est présentée dans la sous-section suivante.

Prenant l'exemple de la question « *Quel est le nom du matériau qui possède la propriété Brillant ?* », nous obtenons la partie *select* et la requête complète.

quel –> *O . select*
nom –> *A . materiau . nom*

SELECT MATERIAU . NOM

SELECT MATERIAU . NOM
FROM MATERIAU, PROPRIETE, QUALIFICATION
WHERE QUALIFICATION . ref_mat = MATERIAU . id_mat
AND QUALIFICATION . ref_prop = PROPRIETE . id_prop
AND PROPRIETE . nom = '*Brillant* '

4.4.3.2. Interface en langage naturel du système QR

Nous avons créé une interface pour le module d'acquisition des connaissances Elle est réalisée en PHP, HTML et JavaScript, des outils décrits dans le chapitre 3.

Cette interface accepte l'introduction des questions en langue française dans un langage naturel et permet de soumettre aisément la question au module du système de question-réponse. Si l'interprétation de la question est bien réalisée, l'interface affiche la réponse. En cas contraire l'utilisateur est conseillé d'introduire de nouveau la question en ajoutant des informations supplémentaires.

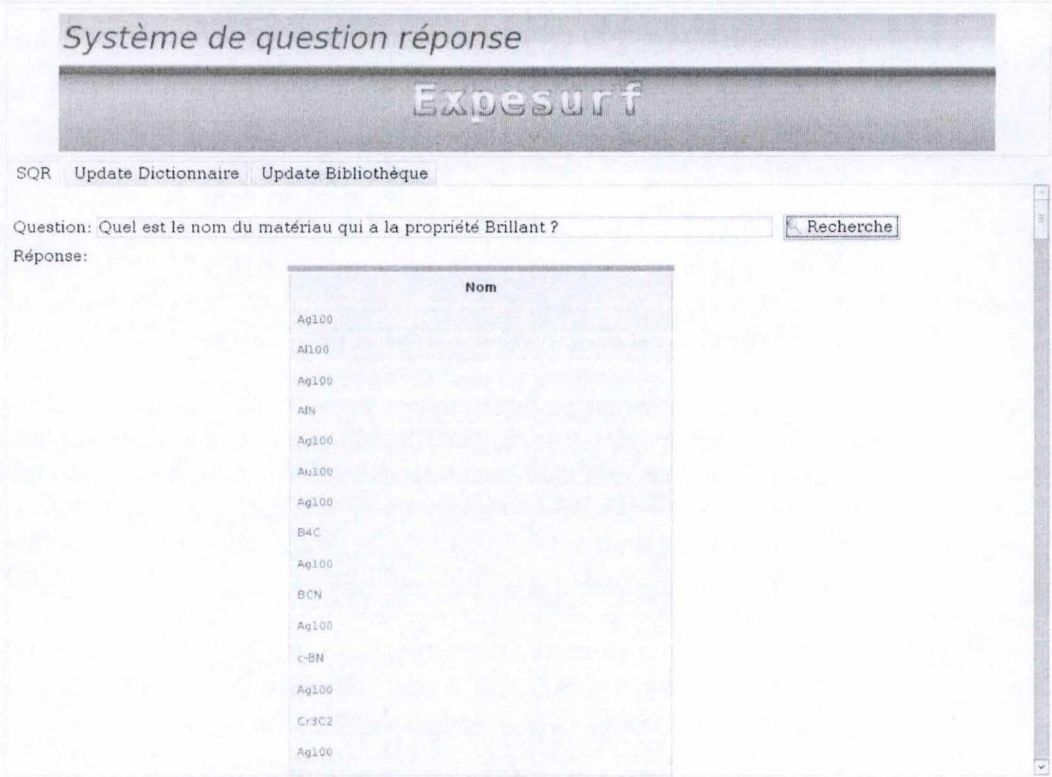


Figure 38 : Interface pour le système de question-réponse EXPESURF

Le lien avec le jar de l'application de traitement de la question se réalise par la commande :

```
<?php
     exec ('java -jar memoire.jar "'.$question.'"', $requête );
?>
```

Figure 39 : Commande PHP qui réalise la liaison interface- système QR

L'interface contient trois onglets : l'un où l'utilisateur pose la question et consulte la réponse et les autres deux qui offrent la possibilité d'alimenter le dictionnaire de mots clés et la bibliothèque des opérateurs avec les données des fichiers XML.

Après ce survol du système d'interrogation, les points 4.4.4. et 4.4.5. présentent les modules au cœur de l'architecture : le dictionnaire et la bibliothèque d'opérateurs.

4.4.4. Le dictionnaire

Le dictionnaire contient des mots clés importants pour une bonne l'interprétation de la question. Un mot clé est un mot pertinent de la question qui peut être :

- un nom de table ou d'attribut ;
- un synonyme ;
- un mot clef pour l'interrogation de la base de données ;
- un opérateur.

Un mot retourné par le dictionnaire a l'une des structures suivantes:

Type . Table . Attribut . Instance
où Type est 'T', 'A' ou 'I'
Attribut et Instance sont optionnels
Type . Operateur
où le Type est 'O'

Où *le type* représente le type du mot clé, *la table* signifie le nom de la table associée, *l'attribut* symbolise le nom de l'attribut de la table et *l'instance* corresponde à la valeur existante dans une table, *l'opérateur* est associé au nom d'un opérateur. La Figure 40 définie la signification de chaque type d'un mot clé et leur structure.

Type	Signification	Structure du mot clé
T	Le mot clé est le nom d'une table	T . Table
A	Le mot clé est le nom d'un attribut	A . Table . Attribut
I	Le mot clé est une instance d'une table	I . Table . Attribut . Instance
O	Le mot clé est associé à un opérateur	O . Operateur

Figure 40 : La signification du type du mot clé

Les informations du dictionnaire sont stockées dans la table *Dictionnaire* de la base de données du logiciel EXPESURF. Elle est décrite dans la Table 19.

Table 19: Table Dictionnaire

Dictionnaire
id_dict
type
table[0-1]
attribut [0-1]
instance[0-1]
opérateur [0-1]

Chaque mot du dictionnaire est identifié par un numéro unique, *id_dict*. Un mot clé peut être une information concernant une donnée de la base de données EXPESURF (table, attribut ou instance d'une table) ou un opérateur (soit mot interrogatif qui indique des informations qui font référence à la construction de la requête, soit des opérateurs relationnels de la base de données). En fonction du type du mot, certains attributs de la table dictionnaire sont optionnels.

Ensuite nous présentons des exemples pour chaque type du mot du dictionnaire:

Exemples

T . materiau
A . materiau . nom
I . propriete . nom . Brillant
O . select

Désambigüisation des attributs

Dans le cas du logiciel EXPESURF nous avons le même nom d'attribut qui apparait dans plusieurs tables. Par exemple le nom peut être le nom d'un matériau, le nom d'une propriété, le nom du test, etc. Pour résoudre cette ambigüité nous utilisons une analyse syntaxique du génitif : le nom génitif (*matériau, propriété, test*) suivi le nom que le détermine (*nom*). Donc pour déterminer l'attribut nous sélectons le code du type « A » qui a la même table avec la table du code suivant.

Par exemple dans la construction « nom du matériau », le dictionnaire va retourne pour le mot clé *nom* une liste codes et pour le matériau T . materiau.

nom du matériau

nom : *A . materiau . nom , A . propriete . nom, A . test . nom, A . procede . nom*
matériau : *T . materiau*

En utilisant ce raisonnement, l'application sélecte le code A . materiau . nom.

4.4.5. Les opérateurs

Un opérateur est un prédicat qui fournit des informations supplémentaires concernant la construction de la requête d'interrogation de la base de données. Son rôle principal est de créer des liaisons entre les mots de la question. Pour la construction de la bibliothèque des opérateurs, nous avons consulté le schéma conceptuel de la base de données et nous avons analysé les mots d'une phrase interrogative qui donnent des informations importantes concernant le type de réponse.

Nous avons considéré comme opérateurs :

- les noms de tables de la base de données ;

- les notions de comparaisons (plus, moins, supérieur, inférieur, maximum, minimum) ;
- les éléments de négation (ne, pas, aucun) ;
- les éléments qui décrivent l'association (et);
- les groupes numériques ou les notions de quantité (combien) .

La structure d'un opérateur est décrite au-dessous :

OPERATEUR (ARG1, ARG2, ARG3)

Où OPERATEUR représente le nom de l'opérateur et ARG1, ARG2, ARG3 sont des éléments de type du code retourne par le dictionnaire. Un opérateur peut avoir un ou maximum trois arguments, donc les ARG2 et ARG3 sont optionnels.

Les opérateurs sont stockés dans la table *Operateur* de la base de données EXPESURF. Elle est décrite dans la Table 20.

Table 20: Table Operateur

Operateur
id_op
nom
arg1
arg2 [0-1]
arg3 [0-1]

Chaque opérateur est identifié par un numéro unique, *id_op*. Un opérateur est caractérisé par un nom et une liste des arguments. Les arguments sont des noms de tables, des attributs ou des instances de la base de données. Un opérateur doit avoir minimum un argument, les autres deux étant optionnels.

Dans notre application nous avons considéré plusieurs types des opérateurs :

∞ les opérateurs sur les tables

Les opérateurs de la base de données sont les opérateurs qui font la liaison entre deux ou trois tables. Ils sont associés avec les tables qui contiennent les références des autres tables. Par exemple nous connaissons que la table *Matériau* et la table *Propriété* sont liées par l'intermède de la table *Qualification*. Nous avons défini l'opérateur *qualification* qui a deux paramètres : matériau et propriété.

qualification (materiau, propriete)

Ces types des opérateurs construisent deux clauses de la requête SQL :

⇨ FROM

Chaque opérateur définit une liste des tables qui est utilisée pour la création de la partie where.

FROM table1, table2, table3

⇨ WHERE

Un opérateur définit une série des conditions qui mettent en évidence la liaison des tables :

WHERE operateur.ref_table1 = table1.id_table1
AND operateur.ref_table2 = table2.id_table2

AND operateur.ref_table3 = table2.id_table2

En consultant le schéma de la base de données nous avons identifié les suivants opérateurs : qualification, quantification, évaluation, solution, application, spécialisation.

∞ les opérateurs pour les instances

Dans le cas où le mot retourné par la recherche du dictionnaire est une instance, nous l'associons à l'opérateur qui a le même nom avec la table à laquelle il appartient. L'opérateur a la structure suivante :

Table (I . Table . Attribut . Instance)

Cet opérateur définit les clauses :

⇨ *FROM table*

⇨ *WHERE A . table . attribut = I . Table . Attribut . Instance*

∞ l'opérateur MAX / MIN

Pour exprimer la forme superlative d'un attribut de type numérique, nous introduisons l'opérateur MAX, respectif MIN. Par exemple pour sélecter le matériau qui a la plus petite température d'utilisation, nous utilisons l'opérateur MIN de forme :

MIN (A . materiau . temp_util)

qui va ajouter dans l'instruction SELECT la fonction SQL *min* :

⇨ *SELECT MIN (A . materiau . temp_util)*

∞ l'opérateur SUP / INF

Les opérateurs SUP et INF sont utilisés pour décrire des notions de comparaison. Ils ont deux arguments : l'une représente un attribut d'une table et l'autre une valeur numérique. Par exemple si nous voudrions sélecter les températures de fusion plus grande de 50, l'opérateur a la forme suivante :

SUP (A . materiau . temp_fusion, 50)

Et définie les clauses :

⇨ *FROM materiau*

⇨ *WHERE A . materiau . temp_fusion > 50*

∞ l'opérateur COUNT

L'opérateur COUNT est utilisé dans les questions dont le mot interrogatif est l'adverbe interrogatif de quantité « combien ». Il contient un seul paramètre de type attribut d'une table de la base de données :

COUNT (A . Table . Attribut)

Ce type d'opérateur ne définit pas des clauses *from* ou *where*, mais ajoute une information importante dans la requête SQL :

⇨ *SELECT COUNT (A . Table . Attribut)*

∞ l'opérateur ET

L'opérateur ET est utilisé pour décrire l'association entre deux attributs. Cet opérateur se traduit dans la requête SQL de façon différente. S'il s'agit des attributs qui apparaissent dans l'instruction SELECT ils sont délimités par virgule (« , »), tandis que dans la clause WHERE est utilisé l'opérateur SQL « AND».

∞ l'opérateur NEGATION

L'opérateur NEGATION est utilisé pour mettre en évidence la négation d'une question. Il corresponde aux mots « ne », « pas », « aucun », « ni » et influence la clause WHERE de la requête, où les conditions sont changées.

∞ l'opérateur SELECT

L'opérateur SELECT est associé à l'instruction SELECT de la base de données. Il a comme paramètres les types de réponse de la question.

SELECT (A . Table . Attribut)

Cet opérateur ne fournit pas des informations pour les clauses *from* ou *where*, mais définit l'instruction select :

⇨ *SELECT A . Table . Attribut*

4.4.6. Compléments sur l'exemple

Ensuite nous présentons une impression d'écran du milieu de travail qui affiche les résultats intermédiaires du système pour la question : « *Quel est le nom du matériau qui possède la propriété Brillant ?* »

```
<terminated> Main [Java Application] /usr/lib/jvm/java-6-openjdk/bin/java (Aug 28, 2011 4:45:25 PM)
Question: Quel est le nom du matériau qui a la propriété Brillant ?

Mots de la question: Quel, est, le, nom, du, matériau, qui, a, la, propriété, Brillant, ?,

Definitions des mots-cles: D.select, A.materiau.nom, T.materiau, T.propriete, I.propriete.nom.Brillant

Operateurs:
qualification(materiau, propriete)
    where: qualification.ref_mat = materiau.id_mat AND qualification.ref_prop = propriete.id_prop
propriete(I.propriete.nom.Brillant)
    where: propriete.nom = 'Brillant';

Requete:
 SELECT DISTINCT materiau.nom
 FROM qualification, materiau, propriete
 WHERE qualification.ref_mat = materiau.id_mat AND qualification.ref_prop = propriete.id_prop
    AND propriete.nom = 'Brillant';

Reponse:
 Ag100
 Al100
 AlN
 Au100
 B4C
 BCN
```

Figure 41: Les résultats intermédiaires du système

On observe dans la Figure 41 que le système parcourt les étapes présentées dans la sous-section 4.4.3.1. et extrait de la base de données plusieurs résultats qui corresponde aux réponses cherchées.

En réalité, dans l'exécution de l'application le système cache toutes ces étapes intermédiaires et envoie vers l'interface seulement la requête construite. À sa place l'interface réalise une consultation de la base de données et affiche la réponse (Figure 38).

L'annexe F présente exemples des questions de test qui peuvent être réalisées pour la consultation de la base de données d'un domaine d'application du traitement de la surface des matériaux, en particulier pour le système expert EXPESURF.

4.4.7. Limitations

Notre système de question-réponse impose quelques limitations du langage naturel :

- Nous utilisons seulement de phrase interrogative partielle qui contient au moins un mot interrogatif.

- Les instances formées par plusieurs mots doivent être introduites comme une citation (entre les « ») :
  Par exemple :
  « Quelles sont les propriétés du matériau « Fer unitaire » ?».

- Dans le cas où il existe de mots absents du dictionnaire, l'utilisateur est conseillé d' ajouter des informations supplémentaires dans la question ou en fonction des mots pertinents déjà introduits, l'application affiche une réponse incomplète.

- Dans le cas ou on n'a pas une interrogation, mais une affirmation, l'application affiche un message conseillé d'introduire de nouveau la question en ajoutant des informations supplémentaires.
  Par exemple : « matériau Cuivre »

Au moment de la rédaction, l'implémentation est toujours en cours. On est en train de réaliser le traitement des suivants types des questions :

- questions complexes formées par plusieurs phases interrogatives
  Quelle est la température de fusion du Cuivre et quelles sont ses propriétés ?

- questions comparatives
  Quels matériaux ont la température de fusion plus grande que 1000 ?

- questions gérant les interactions

Question	Exemple
Quel est le nom de l'interaction qui existe entre le matériau M1 et matériau M2 ?	M1 = Au100 M2 = Zn100
Quels matériaux interagissent de point de vue de l'interaction I?	I = compatibilité galvanique

- questions gérant les incompatibilités

Question	Exemple
Quel procédé est incompatible avec le matériau M1 et matériau M2 ?	M1 = Fe M2 = Cuivre
Quelles sont les catégories pour les incompatibilités ?	

4.4.8. Portabilité

Notre application a été développée sur une machine virtuelle. Elle permet une portabilité facile vers l'autre machine sans être besoin d'une installation des logiciels utilisés. Le DVD annexé au mémoire contient aussi la machine virtuelle avec l'application.

La portabilité du système de question-réponse vers autres logiciels

Le système de question-réponse de l'application a été développé de telle façon qu'il puisse être utilisé par n'importe quel logiciel qui contient des données appartenant à un domaine restreint. Notre système a été développé en Java, un langage qui permet une bonne portabilité. Il sera capable de fonctionner sur des machines ayant des systèmes d'exploitation différents. En plus les informations concernant le domaine spécifique du logiciel EXPESURF (les données de la BD, le dictionnaire ou la bibliothèque des opérateurs, etc.) n'ont pas été introduites dans le code, mais stockées dans des fichiers XML. Ces fichiers XML sont transformés en données de la BD au début de l'application ou peuvent être mis à jour par l'accès à l'onglet « Mis à jour lexique » de l'interface.

Comme tous les systèmes de question-réponse qui utilise comme source de connaissance une base de données, les applications qui veulent utiliser notre plug-in ont besoin d'une interaction directe avec l'administrateur de la BD. Ceci aide en principal à la construction du fichier XML du dictionnaire et de la bibliothèque des opérateurs. Le standard officiel W3C, World Wide Web Consortium, précise qu'un document XML est défini par l'intermède d'un document DTD (Document Type Definition) qui contient le bloc de construction d'un document XML. Ensuite, nous présentons les fichiers DTD utilisés dans notre application.

dictionnaire.DTD

```
<!DOCTYPE dictionnaire [
        <!ELEMENT dictionnaire (mot*)>
        <!ELEMENT mot (nom, type ,table?, attribut?, instance?, operateur?) >
        <!ELEMENT type (#PCDATA)>
        <!ELEMENT table (#PCDATA)>
        <!ELEMENT attribut (#PCDATA)>
        <!ELEMENT instance (#PCDATA)>
        <!ELEMENT operateur (#PCDATA)>
]>
```

Figure 42 : Dictionnaire.DTD

Le document « dictionnaire.dtd » décrit la structure du document « dictionnaire.xml » qui contient le lexique du domaine d'application EXPESURF. L'élément *dictionnaire* est formé par plusieurs mots. Un *mot* contient plusieurs éléments : le nom, le type, le nom de la table, le nom de l'attribut, la valeur de l'instance ou le nom de l'opérateur. Ces éléments sont des chaînes de caractères(« PCDATA ») et certains d'entre eux sont optionnels.

operateurs.DTD

```
<!DOCTYPE operateurs [
    <!ELEMENT operateurs ( operateur* )>
    <!ELEMENT operateur (nom, arg1, arg2?, arg3?) >
    <!ELEMENT nom (#PCDATA)>
    <!ELEMENT arg1 (#PCDATA)>
    <!ELEMENT arg2 (#PCDATA)>
    <!ELEMENT arg3 (#PCDATA)>
]>
```

Figure 43 : Operateur.DTD

La Figure 43 présente le document « operateurs.dtd ». Ce document décrit la structure du document « operateurs.xml » qui définit les opérateurs utilisés dans l'application. L'élément *opérateurs* est formé par plusieurs éléments opérateur. Chaque élément *opérateur* contient un nom et trois arguments qui représentent les noms des tables de la base de données. Ces éléments sont des chaînes de caractères(« PCDATA »).

identifiants.DTD

```
<!DOCTYPE identifiants [
    <!ELEMENT identifiants ( identifiant* )>
    <!ELEMENT identifiant (table, primary, foreign?) >
    <!ELEMENT table (#PCDATA)>
    <!ELEMENT primary (#PCDATA)>
    <!ELEMENT foreign (#PCDATA)>
]>
```

Figure 44 : Identifiants.DTD

La Figure 44 décrit la structure du document « identifiant.dtd ». Ce document est créé pour définir quelles sont la clé primaire (primary key) et la clé secondaire (foregn key) de chaque table de la base de données. Ces informations servent a la création automatique de la clause *where* du chaque opérateur.

Liens_tables.DTD

```
<!DOCTYPE liens [
    <!ELEMENT liens ( lien* )>
    <!ELEMENT lien (table, table1,table2 ?, table3 ?, table4 ?) >
    <!ELEMENT table (#PCDATA)>
    <!ELEMENT table1 (#PCDATA)>
    <!ELEMENT table2 (#PCDATA)>
    <!ELEMENT table3 (#PCDATA)>
    <!ELEMENT table4 (#PCDATA)>
]>
```

Figure 45 : Liens_tables.DTD

La Figure 45 présente un autre fichier DTD qui définit la structure du document Liens_tables.XML. Ce fichier est utilisé dans notre application pour définir les interactions des tables de la base de données. On a choisi d'utiliser ces informations pour améliorer les résultats de recherche des instances dans le cas ou n'est pas précisé le type de l'instance dans la question.

Afin de pouvoir paramétrer la connexion à la base de données sans devoir modifier le code source, on a utilisé un fichier *properties* (BD.properties). Ce fichier contient l'URL, le nom de l'utilisateur et le mot de passe de la base de données.

Dans cette sous-section on a présenté les documents qui doivent être modifiés pour utiliser notre système de question-réponse sur un autre logiciel. Ensuite nous présenterons une courte description de la documentation du système.

4.4.9. Documentation

Le DVD qui est annexé au mémoire contient aussi la documentation de l'application. Nous avons choisi d'utiliser JavaDoc, le générateur de la documentation de la part de Sun Microsystems, grâce à ses avantages :
- est compréhensible par une personne qui ne connaît pas l'application ;
- fait le code plus lisible ;
- est généré de façon automatique en format HTML ;
- contient une description complète des méthodes du projet ;

Un commentaire de la Javadoc contient :
- la description de la méthode (toujours la première ligne) ;
- les paramètres (@param) ;
- le type des réponses (@return) ;
- les exceptions (@throws)

4.5. Conclusion

Dans ce chapitre nous avons présenté une étude de cas des systèmes de question-réponse appliqué au logiciel EXPESURF. Notre recherche se focalise sur l'implémentation d'un système de consultation en langage naturel pour le processus d'Acquisition de Connaissances de notre système expert. En consultant le rapport scientifique du logiciel EXPESURF, [Rapport 3 EXPESURF] nous avons présenté le domaine de l'application et la structure de la base de données.

Le domaine de notre logiciel spécialisé dans le traitement des surfaces des matériaux est très précis et restreint. Dans ce contexte notre approche se base sur une analyse approfondie de la syntaxe des phrases interrogatives de la langue française et sur l'implémentation de la méthode de type *shallow*. C'est une alternative à la construction d'un arbre syntaxique et au traitement sémantique et pragmatique de la question. La grammaire de la phrase interrogative a été décrite en consultant la rubrique de l'interrogation du site des « Études littéraires » [45].

Le développement de notre application se structure en deux outils : un système de question-réponse proprement dit qui réalise le traitement de la question et une interface qui permet d'introduire les questions et d'afficher les réponses. Pour obtenir des réponses correctes et précises aux questions, nous utilisons un dictionnaire des mots-clés et une bibliothèque des opérateurs, solution qui trouve son origine dans l'article « Une bibliothèque d'opérateurs linguistiques pour la consultation de base de données en langue naturelle » écrit par Béatrice Bouchou et al. ([BOUCHOU, 1999]). Le but du dictionnaire est de réaliser une reconnaissance des mots clés de la question, tandis que le rôle des opérateurs est de créer des liaisons entre les mots de la question et de définir la structure des clauses de la requête SQL.

Ensuite nous avons présenté un cas de test pour la question « *Quel est le nom du matériau qui possède la propriété Brillant ?* » en mettant en évidences les étapes de l'application : l'interprétation de la question, l'extraction de mots-clés, la reconnaissance des mots du dictionnaire, la sélection d'opérateurs, la construction de la requête. À la fin de ce chapitre, nous avons décrit la portabilité du système. Elle est assurée par l'utilisation des fichiers XML qui facilite la modification du lexique en fonction du domaine de l'application.

L'implémentation de notre application n'est pas finie. Nous sommes en cours développant du traitement des questions complexes formées de plusieurs propositions et des instances formées par plusieurs mots. L'application est en phase expérimentale et suit à être testée par les experts EXPESURF

Chapitre 5

Conclusion

Dans ce mémoire nous avons abordé le traitement automatique du langage naturel, en particulière les systèmes de question-réponse. Nous avons étudié d'une part les domaines qui exploitent les interfaces en langage naturel et d'autre parte les niveaux de traitement du langage naturel et les éléments techniques de l'élaboration d'un système de question-réponse.

En comparaison avec les autres travaux qui réalise une consultation en langage naturel de la base de données, l'approche que nous l'avons proposé s'appuie sur une analyse approfondie de la syntaxe des phrases interrogatives de la langue française et de la structure de base de données. C'est une alternative au traitement sémantique et pragmatique de la question et à la construction d'un arbre syntaxique (méthode utilisée dans Chat-80, [PEREIRA, 1983]). Dans notre développement nous gardons l'idée de l'article [BOUCHOU, 1999] : de traduire la question en langage naturel vers une requête SQL par l'intermédiaire d'un dictionnaire de mots clefs et d'une bibliothèque des opérateurs.

L'application que nous avons élaborée en vue de la présentation de notre mémoire de fin d'études en sciences informatiques consiste à développer et à mettre en œuvre un système de question-réponse pour le logiciel EXPESURF. Ce logiciel est un système expert dans le domaine de surface engineering qui fournie des informations concernant le choix des traitements des surfaces des matériaux à appliquer sur une nouvelle pièce et l'amélioration des propriétés selon les besoins.

Dans le cadre du logiciel EXPESURF le problème est plus spécifique : le monde de traitement de la surface des matériaux est un domaine restreint, très précis et peu variable. Notre travail est particulier, étant limité en rapport avec le langage naturel complexe. Pour ces raisons l'analyse approfondie de la syntaxe de la question est suffisamment.

Notre système question-réponse implémenté dans le processus d'Acquisition de Connaissances du logiciel EXPESURF amène plusieurs fonctionnalités :

- ⇨ la réalisation facile du multitraitement .
- ⇨ le traitement d'une grande quantité d'informations dans un délai plus court ;
- ⇨ l'ouverture générale vers les clients qui n'ont pas une formation spécifique ;

Travaux futurs

Pour améliorer les performances de notre système de consultation en langage naturel de la base de données nous citons quelques propositions de perspectives :

- Ajouter un correcteur automatique pour la question pour éviter l'existence des erreurs d'orthographe.

- Utiliser une ontologie pour une meilleure analyse sémantique de la question et pour remédier les problèmes d'ambiguïté.

- Enrichir la base de connaissances du dictionnaire avec plus de synonymes dans le but de réaliser une meilleure reconnaissance des mots clés.

- L'implémentation des clauses : Group by et Order by
  Exemple :
  Quels sont les premiers 10 matériaux avec la plus grande dureté ?
  Quels sont les matériaux qui ont la propriété brillant et sont ordonnés croissant selon la température d'utilisation ?

BIBLIOGRAPHIE

1. **[AUDIBERT, 2009]** AUDIBERT, L. Base de données de la modélisation au SQL, Université Paris 13 – Laboratoire d'Informatique de Paris-Nord (LIPN), 2009

2. **[AUDIBERT, 2010]** AUDIBERT, L. Traitement automatique du langage naturel (Outils d'analyse de données textuels), Université Paris 13 – Laboratoire d'Informatique de Paris-Nord (LIPN), 2010

3. **[BAUZON, 2009]** BAUZON, M. L'essor du langage Java, Journal du Net, 2009, URL : http://www.journaldunet.com/developpeur/expert/java-j2ee /38746/l-essor-du-langage-java.shtml, (consulté le 21/07/2011)

4. **[BOUCHOU, 1999]** BOUCHOU B., MAUREL D. Une bibliothèque d'opérateurs linguistiques pour la consultation de base de données en langue naturelle, Tours, 1999 URL : http://www.atala.org/doc/actes_taln/AC_0014.pdf (consulté le 28/08/2011)

5. **[BOULOGNE, 2004]** BOULOGNE, A. Le Vocabulaire de la documentation , ADBS (Association des professionnels de l'information et de la documentation), Paris, 2004

6. **[BURAGA, 2002]** BURAGA, S. et al., Programare Web în bash şi Perl, Ed. Polirom, Iaşi, 2002

7. **[CARSTOIU, 1994]** CARSTOIU, D. I. Sisteme expert, 1994, Editure ALL, Bucarest

8. **[CHERAGUI, 2010]** CHERAGUI, M. A. Un modèle d'analyse multicritère de la levée de l'ambiguïté associé à un Tagger pour le Traitement Automatique de l'arabe, 2009/2010

9. **[CHOMSKY, 1957]** CHOMSKY N. , Syntactic Structures, 1957

10. **[CREPS]** CREPS, Centre de Documentation et d'Information, Montpellier, URL : http://www.creps-montpellier.org (consulté le 21/07/2011)

11. **[DAUBEY, 2010]** DAUBEY B., Traduction humaine et traduction automatique 7 juin 2010, Journal de la Traduction URL : http://blog.atenao.com/traduction-professionnelle/traduction-humaine-et-traduction-automatique-128 (consulté le 7/07/2011)

12. **[EL-BEZE, 2006]** EL-BÈZE, M. Systèmes de question-réponse URL : http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/ fich_art/712-3-3-SystQ-R.pdf, LIA, 2006, 15-19, (consulté le 7/07/2011)

13. **[EL AYARI, 2007]** EL AYARI, S. Evaluation transparente de systèmes de questions-réponses :application au focus, Recital 2007, Toulouse, URL : http://www.limsi.fr/~sarra/Publications/EvalFocus.pdf (consulté le 8/07/2011)

14. [GENTHON, 2004] GENTHON P., L'intelligence artificielle, Toulouse, 2004
URL : http://cnam.toulouse.free.fr/cnam/19900-ia_2004.pdf
(consulté le 20.08.2011)

15. [GIONEA, 2008] GIONEA, I. Inteligenta artificiala,
URL :http://www.catia.ro/articole/ai/ai.htm, 2008,
(consulté le 5.07.2011)

16. [GRAU et al., 2005] GRAU, B., MAGRININI B. Systèmes de question/réponse, ATALA, Vol.46, Nr. 3, 2005
URL : http://www.atala.org/Systemes-de-question-reponse,
(consulté le 7.07.2011)

17. [HARMACH et al., 2003] HARMACH, S., MAHE, S., MARIEY, H. TE :Application de Prolog en IA en particulier les systèmes experts, 2003
URL :
http://deptinfo.unice.fr/twiki/pub/Linfo/PlanningDesSoutenances2002/HARMACH-MAHE-MARIEY.pdf (consulté le 20.07.2011)

18. [HRISTEA, 2000] HRISTEA, F., Introducere in procesarea limbajului natural cu aplicatii in Prolog, Ed. Universitatii Bucuresti, Bucarest, 2000

19. [HUANG, 2006] Huang T.-M., Kecman V., Kopriva I. *Kernel Based Algorithms for Mining Huge Data Sets, Supervised, Semi-supervised, and Unsupervised Learning*, Springer-Verlag, Berlin, 2006

20. [IVASCU, 2005] IVASCU, V. Introducere in PHP si MySQL, 2005
URL : http://www.oriceon.com/, (consulté le 18.07.2011)

21. [KATZ, 1997] KATZ,B. Annotating the World Wide Web Using Natural Language, 1997

22. [L'HAIRE, 2000] L'haire, S. L'enseignement assisté par ordinateur et le traitement du langage naturel, Université de Genève – Faculté des Lettres, 2000
URL : http://sebastien.lhaire.org/publis/lhairedeslight.pdf,
(consulté le 17.07.2011)

23. [MORICEAU, 2009] MORICEAU, V. Les systèmes de question – réponse, Université Paris-Sud, 2009,
URL :http://www.limsi.fr/Individu/anne/DEA/QR_cours_m2.pdf
(consulté le 1/07/2011)

24. [RAO et al., 2010] RAO G. et al., Natural language query processing using semantic grammar / (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 219-223

25. [Rapport 3 EXPESURF] EXPESURF - Aide à l'ingénierie de surface multitraitement par système expert modulable, Rapport technique annuel, No. 3, 2007, p.108-122

26. [Rapport 4 EXPESURF] EXPESURF - Aide à l'ingénierie de surface multitraitement par système expert modulable, Rapport technique annuel, No. 4, 2008, p.19

27. [ROZENKNOP, 2010] ROZENKNOP, A. Modèles de Langage et Analyse Syntaxique, Université Paris 13 – Laboratoire d'Informatique de Paris-Nord (LIPN), 2010
URL : http://www-lipn.univ-paris13.fr/~rozenknop/Cours/ITCN_MLAS/Seance1/Cours.article.pdf (consulté le 20/08/2011)

28. [NEWELL, 1982] NEWELL, A. The Knowledge Level, *Artificial Intelligence*, 1982

29. [NIE, 2008] NIE, J.-I. Recherche d'Information, *Introduction*, Montréal, 2008
URL : http://www.iro.umontreal.ca/~nie/IFT6255/Introduction.html (consulté le 8/07/2011)

30. [PEREIRA, 1983] PEREIRA, F. Logic for natural language analysis, University of Edinburgh, 1983
URL: http://www.ai.sri.com/pubs/files/669.pdf
(consulté le 2/08/2011)

31. [PISTOL, 2011] PISTOL, I. Invatare automata, Universitatea Alexandru Ioan Cuza, Iasi, 2011
URL : http://profs.info.uaic.ro/~ipistol/ia1011/res/IA.pdf
(consulté le 15/08/2011)

32. [SPERBER, 2007] SPERBER, M. et al. The Revised[6] Report on the Algorithmic Language Scheme, 2007
URL: http://www.r6rs.org/, (consulté le 25/08/2011)

33. [TADEUSZ, 1999] TADEUSZ, B. , TADEUSZ, W. Surface Engineering of Metals , 1999

34. [TELLIER, 2010] TELLIER, I. Introduction au TALN et à l'ingénierie linguistique,2010
URL :http://www.univ-orleans.fr/lifo/Members/Isabelle.Tellier /poly_info_ling/info-ling.pdf, (consulté le 1/07/2011)

35. [TRAUSAN, 2004] TRAUSAN, S.M. Inteligenta artificiala si web semantic
ISBN 973-681-682-6, Ed. Polirom, 2004

36. [TUFIS, 2008] TUFIS, D. Sisteme de întrebare-răspuns în limbaj natural pentru spații de căutare deschise, Bucureşti, 2008, p.17, URL :
http://dtil.unilat.org/seminar_bucuresti_2008/actes/Tufis%20Dan.pdf
(consulté le 7/07/2011)

37. [VASILESCU, 1996] VASILESCU, R. Inteligenta Artificiala, 1996,
URL :http://www.cs.cmu.edu/~mihaib/articles/ai/ai-html.html
(consulté le 2/07/2011)

38. [Wiki, fr] « Systèmes de questions-réponses »
http://fr.wikipedia.org/ (consulté le 6/07/2011)

39. [YVON, 2007] F. YVON, Une petite introduction au traitement Automatique du langage naturel, support de cours, Ecole Nationale Supérieur des télécommunications, 2007,
URL :http://www.univ-orleans.fr/lifo/Members/Isabelle.Tellier/,
(consulté le 25/06/2011)

GLOSSAIRE DES ACRONYMES

AC	Processeur Acquisition de Connaissances du logiciel EXPESURF
AD	Logiciel Acquisition de Données du logiciel EXPESURF
ALPAGE	Analyse Linguistique Profonde A Grande Echelle, projet au Centre de Recherche INRIA Paris-Rocquencourt, dont le domaine de recherche est le Traitement Automatique des Langues
ATALA	Association pour le Traitement Automatique des Langues
CHAT	Système de question-réponse dans le domaine de la géographie du monde, implémenté en Prolog
CLEF	Cross Language Evaluation Forum, campagne d'évaluation pour les systèmes de question-réponse multilingue
DyALOG	Un environnement pour construire des analyseurs syntaxiques tabulaires pour les langues naturelles
DTD	Document Type Definition
ELIZA	Système de question-réponse qui simule le comportement d'un psychologue
EQUER	Évaluation en Question Réponse, campagne d'évaluation pour les systèmes de question-réponse en français
EXPESURF	System expert spécialise dans le traitement de la surface des matériaux
FIPS	Un outil d'étiquetage créé par LATL qui développe des analyseurs syntaxiques pour différentes langues
GUI	Graphical User Interface
Haskell	Langage de programmation fonctionnel dérivé du LISP
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
LATL	Laboratoire d'Analyse et de Technologie du Langage
LISP	LISt Processing
LUNAR	Système de question-réponse qui accède des données du domaine chimique correspondantes aux roches trouvées lors des missions Apollo.
MIR	Moyenne de l'Inverse du Rang
MIT	Massachusetts Institute of Technology
MYCIN	Système de question-réponse dans le domaine de médecine
NLP	Natural Language Processing
NTCIR	NII Test Collection for IR Systems, campagne d'évaluation pour les systèmes de question-réponse en japonais et chinois
PERL	Practical Extraction and Report Language
PHP	Hypertext Preprocessor
PostgreSQL	Un système de gestion de base de données utilisé par EXPESURF
PROLOG	PROgramming in LOGic
Système QR	Système de question réponse
QUAERO	Campagne d'évaluation pour les systèmes de question-réponse en français et anglais
RI	Recherche d'information
Scheme	Langage dérivé du LISP
SGBD	Gestion de Base de Données

SGBDR	Gestion de Base de Données Relationnelle
SHRDLU	Système de question-réponse dans le monde des figures géométriques qui modifie un ensemble des cubes
SQL	Structured Query Language
SQR	Système de question-réponse
SYSTRAN	Logiciel de traduction automatique
TALN	Traitement automatique du langage naturel
TREC	Text REtrieval Conference, campagne d'évaluation pour les systèmes de question-réponse en anglais
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

ANNEXES

Annexe A : Liste des figures

Figure 1 : Schéma très simplifié élaboré par des psychologues cognitivistes 9
Figure 2 : Architecture séquentielle de la suite des traitements d'une application du TALN 10
Figure 3 : Les interactions des systèmes des questions-réponses avec autres domaines 16
Figure 4 : Type de séparateurs 21
Figure 5 : Exemple de traitement lexical 22
Figure 6 : Les éléments d'un arbre de dépendance 23
Figure 7 : L'arbre de dépendance pour la question "Quelle est la propriété du matériau Cuivre?" 23
Figure 8 : L'architecture de systèmes de question-réponse 34
Figure 9 : Les types des requêtes 35
Figure 10 : Les types d'objets 35
Figure 11 : Les types de phrases 36
Figure 12 : Recherche des documents dans un système de question-réponse 36
Figure 13 : Exemple d'extraction des mots clés 37
Figure 14 : Exemple de passage candidat et réponse candidate 37
Figure 15 : Exemple de patron d'extraction 38
Figure 16 : Exemple de patron de base et patrons acquis 39
Figure 17 : Relations syntaxique pour la question « Quel métal a le plus haut point de fusion ? » 39
Figure 18 : Arbre de dépendances pour la question « Quel métal a le plus haut point de fusion ? » 40
Figure 19 : Relations syntaxiques à parti de l'arbre de dépendances 40
Figure 20 : Extraction des réponses utilisant de relations syntaxiques 40
Figure 21 : Diagramme des documents 43
Figure 22 : Définition du rappel 43
Figure 23 : Définition de la précision 43
Figure 24 : Courbe de dépendance entre le rappel et la précision 44
Figure 25 : L'environnement Eclipse 52
Figure 26 : Fonctionnement PHP 54
Figure 27 : L'interface Fips 56
Figure 28 : L'analyse syntaxique du Fips 56
Figure 29 : Outils utilisés en EXPESURF 57
Figure 30 : L'architecture d'un système expert 60
Figure 31 : L'architecture du logiciel EXPESURF 61
Figure 32 : Le schéma de la base de données EXPESURF 63
Figure 33 : Les pronoms interrogatifs 73
Figure 34 : Les formes composées du pronom interrogatif 73
Figure 35 : Les adverbes interrogatifs 74
Figure 36 : Les adjectifs interrogatifs 74
Figure 37: Structure du système de question-réponse EXPESURF 76
Figure 38 : Commande PHP qui réalise la liaison interface- système QR 82
Figure 39 : La signification du type du mot clé 83

Figure 40: Les résultats intermédiaires du système 87
Figure 41 : Dictionnaire.DTD 89
Figure 42 : Operateur.DTD 90
Figure 43 : Identifiants.DTD 90
Figure 44 : Liens_tables.DTD 90

Annexe B : Liste des équations

Equation 1 : La définition d'un système expert 11
Équation 2 : Definition du Moyenne de l'Inverse du Rang 42

Annexe C : Liste des tableaux

Table 1: Tableau Matériau 64
Table 2: Table Propriété 64
Table 3: Table Qualification 65
Table 4: Table Test 65
Table 5: Table Quantification 66
Table 6: Table Evaluation 66
Table 7: Table Interaction 66
Table 8: Table Prop_doubles 66
Table 9: Table Procédé 67
Table 10: Table Caractérisation 67
Table 11: Table Spécialisation 67
Table 12: Table Application 67
Table 13: Table Critère 68
Table 14: Table Incompatibilité 68
Table 15: Table Solution 69
Table 16: Table Resolution 69
Table 17: Table Sol_Proc 70
Table 18: Table Sol_Couche 70
Table 19: Table Dictionnaire 83
Table 20: Table Operateur 85

Annexe D : Contenu de DVD

Le DVD joint à notre mémoire contient les répertoires suivants:

- /memoire/ contenant ce mémoire en format Portable Document Format (PDF)
- /machine virtuelle/ contenant la machine virtuelle avec l'application
- /implementation/SQR/src/ contenant les fichiers sources de l'application et les fichiers XML et properties
- /implementation/SQR/doc/ contenant la documentation de l'application
- /implementation/SQR/lib/ contenant les librairies utilisées dans l'application
- /implementation/interface/www contenant les fichiers source de l'interface

Annexe E : Ressources utiles

- Java Runtime Environment:
  http://java.sun.com.javase/downloads/index.jsp ;

- PostgreSQL:
  http://www.postgresql.org/ ;

- JDBC pour PostgreSQL:
  http://jdbc.postgresql.org/ ;

- Apache :
  http://httpd.apache.org/ ;

- PHP:
  http://www.php.net ;

Annexe F : Exemple des questions

Cette annexe présente quelques exemples des questions :

Matériau

Question	Exemple
Quelle est la température de fusion (la conductivité électrique, la température d'utilisation, la description) du matériau M ?	M = Cuivre
Quel matériau a la plus petite (plus grande) température de fusion ?	
Quels matériaux ont la température de fusion plus petite (plus grande) que T ?	T = 1000

Matériau - Propriété

Question	Exemple
Quelles propriétés possède le matériau M ?	M = Cu100
Quels matériaux ont la propriété P ?	P = Brillant
Combien des matériaux ont la propriété P ?	P = Brillant

Matériau – Propriété – Test

Question	Exemple
Pour le matériau M, quelles sont les propriétés évaluées par les tests ?	M = Cu100
Quels matériaux ont la propriété P évaluée par les tests ?	P = Brillant
Quel est le nom du matériau qui a la propriété P évaluée par les tests ?	P = Brillant

Matériau - Test

Question	Exemple
Quels sont les tests qui peuvent être effectués pour le matériau M ?	M = Cu100
Quels matériaux ont été testés pour T ?	T = résistivité électrique

Propriété – Test

Question	Exemple
Quels tests évaluent la propriété P ?	P = Brillant
Quelles sont les propriétés concernées par le test T ?	T = résistivité électrique

Procédé – Critère

Question	Exemple
Quels sont les critères qui caractérisent le procédé Pc ?	Pc = Revêtement
Quels procédés sont caractérisés par le critère C ?	C = Rugosité

Matériau – Procédé

Question	Exemple
Quels sont les procédés qui spécifient le matériau M ?	M = FeB
Pour quels matériaux est appliqué le procédé P ?	P = Traitement thermique à la flamme

Cas spéciaux :

∞ plusieurs éléments à sélecter

Quelles sont la température de fusion et la température d'utilisation du matériau qui a la propriété P ? (e.g. P = Brillant)

∞ plusieurs conditions à accomplir

Quels sont les matériaux qui possèdent la propriété P1 et la propriété P2 ? (e.g. P1 = Brillant , P2 = Base)

∞ le type de l'instance n'est pas précisé

Quelle propriété a M ? (e.g. M = Cu100)

Quels matériaux sont P ? (e.g. P = Brillant)

Annexe G : Outils de TALN

Nom d'outil	Nom d'auteur	Type d'outil
ACABIT	Daille Béatric	Extraction de termes
ANA	Enguehard Chantal	Extraction de termes
Analyseur LFG	Vapillon Jérôme	Analyse syntaxique, Gestion de lexique
Analyseur syntaxique du GREYC	Groupe Syntaxe du GREYC	Analyse syntaxique, Traitement de corpus
Atelier LTAG	Lopez Patrice	Analyse syntaxique
ATLAS SEMANTIQUES/ SEMANTIC ATLAS	Ploux Sabine	Gestion de lexique
Class4U	Garnier Alain	Étiquetage, Traitement de corpus
CooLoX	Audibert Laurent	Traitement de corpus
CORDIAL Universités ou Cordial Analyseur	Laurent Dominique	Étiquetage, Analyse syntaxique, Traitement de corpus, Extraction de termes
CorTeCs	Heiden Serge	Étiquetage
DOSLoX & WinLoX	Audibert Laurent	Traitement de corpus, Extraction de termes
DyALog	Clergerie Eric	Analyse syntaxique
ESSENTIAL	Abderrafih Lehmam	Étiquetage, Traitement de corpus, Gestion de lexique,
FASTER	Jacquemin Christian	Extraction de termes
FipsTag	Wehrli Eric	Étiquetage
Humanistique	Antoni Marie-Hélène	Étiquetage, Traitement de corpus, Gestion de lexique
Incremental Finite State Parser (IFSP)	Salah Ait-Mokhtar, Jean Pierre Chanod	Analyse syntaxique
INTEX	Silberztein Max	Étiquetage, Analyse syntaxique, Traitement de corpus, Extraction de termes, Gestion de lexique
Ips	Wehrli Eric	Analyse syntaxique
Labelgram	Sylviane Cardey, Zahra El Harouchy, Peter Greenfield	Étiquetage
LEXTER	Bourigault Didier	Analyse syntaxique, Extraction de termes
Maatala	Mullon Pascal	Gestion de lexique
Moteur de recherche Noematics	VERNEY Daniel	Traitement de corpus
SyntEtiq	Bidon Hélène	Étiquetage, Analyse syntaxique
Tatoo	Robert Gilbert	Étiquetage
TerminologyExtractor	Cornu Etienne	Extraction de termes
XeLDA	Xerox MKMS	Étiquetage, Analyse syntaxique, Extraction de termes
Xerox POS tagger	XEROX (divers auteurs)	Étiquetage
XLFG	Clément Lionel	Analyse syntaxique

Source: ATALA (http://www.atala.org/-Outils-pour-le-TAL-)