Stratégies d'apprentissage, paraphrases et analyse de texte

par

Pierre PLANTE
Université du Québec, Montréal - CANADA

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

APLEC (l'Apprenti-Lecteur) est une extension du logiciel Déderec, un système de programmation consacré au traitement linguistique et à l'analyse de contenu des textes. (1)

De façon générale, APLEC permet de programmer une confrontation entre une séquence écrite en langage naturel et un texte écrit dans la même langue.

A la fois la séquence et le texte doivent être décrits par une même grammaire. Le travail d'APLEC consistera essentiellement à ramener pour la séquence donnée et eu égard à une programmation donnée de la confrontation, une (ou plus d'une) séquence du texte analysé.

APLEC peut ainsi servir à construire un système questions-réponses, où la séquence analysée est la question, et les séquences dépistées dans le texte la réponse.

APLEC pourra aussi aider à cerner les limites de capacité d'analyse de contenu d'une grammaire donnée. Il pourra être utilisé pour améliorer (c'est de là qu'il tire son nom) ces possibilités d'analyse de contenu d'une grammaire, amélioration ou apprentissage qui se fera en interactivité, au fur et à mesure des conversations.

Enfin, on verra qu'APLEC est à même de pouvoir ramener pour une séquence et une grammaire données les séquences du texte qui ont avec la séquence analysée des relations de type paraphrastique, c'est-à-dire des relations régies par des règles de transformation; ces règles se trouvant constituer (avec d'autres règles) le programme de la confrontation.

On doit fournir à APLEC, outre la séquence et le texte que l'on veut confronter, deux types de données.

D'abord on doit nourrir APLEC de la grammaire que l'on veut voir appliquer sur la séquence et sur le texte. Ici APLEC exige une grammaire de type Déredec. Ces grammaires aussi appelées GDT (Grammaires Descriptives de Texte) ont la forme d'automates à états finis récursifs, à déterminisme contrôlable, et où le balayage contextuel peut tout aussi bien se faire de droite à gauche que de gauche à droite. Différentes opérations du logiciel Déredec sont programmables dans ces automates et permettent de construire des syntagmes sur les éléments des séquences analysées, de relier les nœuds dans ces structures arborescentes par des relations de dépendance contextuelle orientées, et enfin d'associer aux mots des séquences sémantiques de complexité variable.

Les GDT modifient les séquences d'un texte en leur adjoignant des structures descriptives de certaines régularités grammaticales. Le Déredec peut tout aussi bien servir croyons-nous, à l'indexation de caractéristiques morphologiques, syntaxiques, sémantiques voire logiques ... (2).

L'autre donnée (outre donc la GDT) que l'usager doit fournir à APLEC consiste en une liste de modèles d'explorations. Les modèles d'exploration permettent de dépister de façon sélective certaines structures ou certains éléments des descriptions produites par les grammaires.

Illustrons par un exemple, à la fois une GDT et l'application de quelques modèles d'exploration. Nous utiliserons une grammaire dédiée au dépistage des structures syntaxiques de surface du français courant. De façon générale, cette grammaire (Plante octobre 1980) peut dépister pour toute

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

proposition, le thème et le propos (focus/comment), deux types de compléments verbaux (les directs et les circonstantiels), et plusieurs types de déterminations nominales. Elle pratique en parallèle une segmentation de la phrase et de ses principaux constituants syntagmatiques et réussit notamment à ce niveau à séparer les propositions entre elles. Dans cette grammaire nous avons différencié trois types de catégories descriptives : les catégories de base, les catégories temporaires qui servent de relais à l'indexation des catégories de base, et les catégories syntagmatiques qui, comme leur nom l'indique, représentent des regroupements sur les catégories.

L'ensemble de tout le traitement est divisé en deux étapes : dans la première, un dictionnaire et des automates interactifs indexent les catégories de base aux lexèmes, alors que, dans la seconde, d'autres automates composent les groupes syntagmatiques, dépistant parallèlement les relations de dépendance contextuelle. Les catégories de base sont les primitifs de la grammaire implantée; c'est-à-dire qu'une fois leur indexation terminée, toutes les manipulations concernant la composition syntagmatique et le dépistage des relations de dépendance contextuelle sont entiérement automatisées.

Les catégories de base :

N1....	Noms propres et communs;
N211..	Pronoms pers. sujets (je, tu ...);
N212..	Pronoms pers. compléments (te, me ...);
N22...	Autres noms;
N23...	Pronoms relatifs;
D11...	Déterminants verbaux (ne pas ...);
D12...	Déterminants nominaux (mes, ton ...);
D2....	Déterminants connecteurs (que, comme ...);
V1....	Autres verbes;
V21...	Infinitifs;
V22...	Participes présents;
V23...	Participes passés et Adjectifs;
C1....	Conjonctions de coordination;
C21...	Prépositions faibles (de, d);
C22...	Prépositions fortes (avec, sans...);
C31...	Ponctuation faible (,);
C32...	Ponctuations fortes (. ; : ? ...);

Le dictionnaire permet de catégoriser environ 90 % des «tokens» d'un texte (ce pourcentage varie selon la redondance lexicale du texte analysé) ; notre dictionnaire contient actuellement quelque 14.000 lexèmes et s'enrichit à chaque nouvelle application sur un nouveau corpus.

Après le passage du dictionnaire, des automates interactifs permettent de visionner le contexte des mots restants et de fournir «à la main» des catégories de base à ces derniers.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

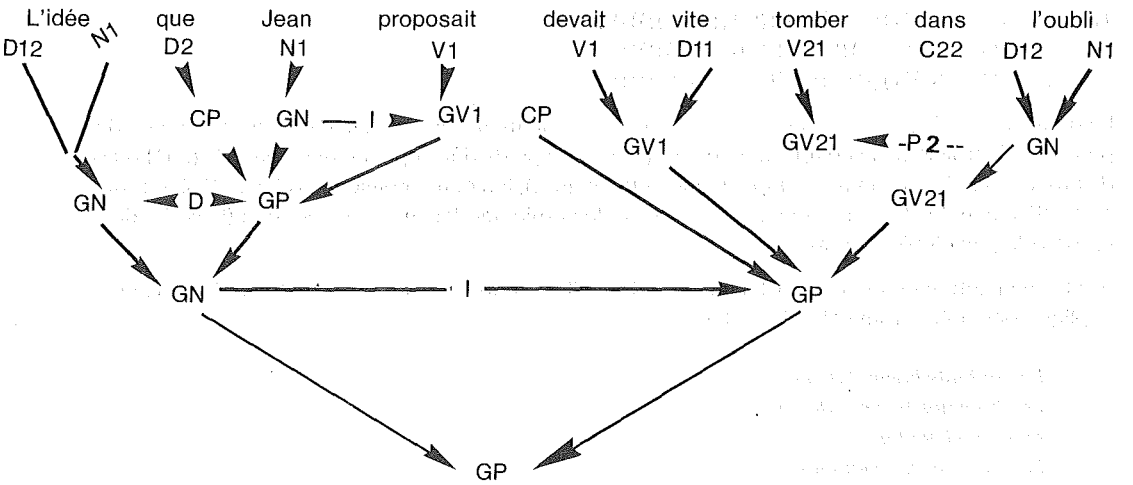
Exemple d'une phrase à ce niveau de sortie :

L' idée que Jean proposait devait vite tomber dans l'oubli.

D12 N1 D2 N1 V1 V1 D11 V21 C22 D12 N1

La seconde étape du traitement est subdivisée en trois temps. Des automates vont d'abord composer sur les catégories de base indexées aux lexèmes, les molécules les plus fortement liées (certains syntagmes déterminatifs et nominaux), puis d'autres automates vont assurer le dépistage des connecteurs propositionnels dans la phrase, et enfin, un dernier groupe d'automates va construire l'ensemble des syntagmes emboîtés.

Séquence décrite par la grammaire de surface :



Les catégories syntagmatiques :

GN Groupe nominal ;
GV1 Groupe verbal construit sur un V1 ;
GV21 Groupe verbal construit sur un V21 ;
GV22 Groupe verbal construit sur un V22 ;
GV23 Groupe verbal construit sur un V23 ;
GD11 Groupe déterminant construit sur un D11 ;
GP Groupe propositionnel ;
CP Connecteur de propositions ;

On aura aussi GNCO, GV1CO, GV21CO... GPCO... pour chacun des syntagmes coordonnés.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

Cinq relations de dépendance contextuelle codifient les rapports possibles entre les catégories du système :

--I-- l'interdépendance ou la relation thème/propos (aussi marquée : TP) ;
--P1-- première relation de développement du propos ;
--P2-- seconde relation de développement du propos ;
--D-- la relation de détermination nominale (aussi marquée : DET) ;
--CO-la relation de coordination.

L'ensemble de notre grammaire totalise une soixantaine d'automates où se répartissent quelque 1.700 règles. Elle a jusqu'à présent été appliquée sur plus d'un million de mots.

Voyons maintenant l'exploration de ce texte par trois modèles d'exploration : (3)

```
THE*...   (GP  ((GV1)(TP -)(GN((N1((= EA)))))) ;
PRO*...   (GP  ((X)(TP +)(GV1((V1((= EA)))))) ;
CP1*...   (GP  ((GV1)(P1 -)(GN((N1((= EA)))))) ;
```

Le premier modèle peut dans une proposition dépister le thème, le deuxième dépistera le verbe du propos, et le troisième le complément direct (premier type de développement du propos (P1)) du verbe du propos. Ainsi dans notre exemple (L'idée que Jean ...), les deux premiers modèles dépisteront respectivement les lexèmes «Jean», «devait», et le dernier modèle ne dépistera rien (le verbe du propos n'a pas d'objet direct).

APLEC a maintenant tout ce qu'il lui faut pour travailler. Supposons un court texte sur lequel nous appliquerons notre grammaire de surface :

Les enfants boivent du lait.
Les hommes boivent du café.
Marie se lève tard.
Les enfants se lèvent tôt.

Nous fournissons de plus à APLEC la liste des modèles entrevus (THE* PRO* CP1*), et sommes prêts à commander une première confrontation entre une séquence et la description de texte (c'est-à-dire le texte une fois ses structures de surface indexées). Les modèles d'exploration activent la comparaison entre la séquence et le texte. Ils sont étudiés un à un dans l'ordre proposé par l'usager. Un modèle donné sert à la fois à dépister un lexème dans la séquence puis par la suite servira à dépister toutes les séquences du texte qui contiennent ce lexème dans la même position structurale (grammaticale) que celle indiquée par le modèle. Par exemple le modèle THE* dépistera le thème de la séquence analysée et toutes les séquences du texte qui ont ce même lexème en position thème.

Exemple de conversation avec APLEC sur notre petit texte :

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

usager : *Les enfants boivent le lait ?*
APLEC : *Les enfants boivent du lait ?*
Autre question ?

Avec le premier modèle (THE*), APLEC a dépisté «enfants» sur la séquence question («enfants» est le thème de la question), puis il a parcouru le texte et a ramené toutes les séquences ayant «enfants» pour thème, à savoir : la première et la dernière phrase du texte. Par la suite il a dépisté «boivent» sur la séquence question à l'aide du deuxième modèle (PRO*), et a fouillé la réponse courante (on appelle ainsi les séquences du texte dépistées par le jeu des modèles déjà étudiés). Seule la première phrase du texte renferme comme dans la question (et dans la bonne position : PRO*) le lexème «boivent», et deviendra de ce fait la nouvelle réponse courante (la deuxième phrase se trouvant du coup éliminée de la réponse courante). Le troisième modèle extrait de la question le lexème «lait», et APLEC constate que ce mot occupe la même position CP1* dans la réponse courante. Cette dernière deviendra la réponse définitive, la liste des modèles étant épuisée.

Si un modèle ne trouve aucun écho sur la réponse courante, APLEC modifie l'ordre des modèles dans la liste et recommence son travail. Le modèle problématique se retrouvant à la queue de la liste, tous les autres modèles se rapprochent d'un cran du début de la liste. Il est à remarquer que seuls les modèles qui dépistent quelque chose sur la question sont traités dans la confrontation. Au début de celle-ci, APLEC teste tous les modèles de la liste proposée par l'usager et en extrait tous ceux qui ne peuvent dépister de lexèmes sur la séquence question. Cette caractéristique permet à l'usager de construire une liste de modèles très générale valable pour plusieurs types de questions ; APLEC épurera à chaque fois la liste par l'analyse particulière de chaque question.

1.0 Première stratégie d'apprentissage

Si APLEC est en mode «apprentissage» et qu'un modèle de la liste ne trouve aucun écho sur la réponse courante, soit parce que le lexème de la question (dépisté par le modèle) n'y apparaît pas, ou soit qu'il n'y apparaît pas dans la position structurale indiquée par le modèle, APLEC interrompt son travail et entre en communication avec l'usager ; il lui demandera de confirmer ou d'infirmer l'existence d'une équivalence sémantique entre le lexème de la question et l'un ou l'autre des lexèmes de la réponse courante occupant la même position structurale (décrite par le modèle courant) que le lexème de la question :

usager : *Marie boit-elle?*
APLEC : *Pouvez-vous fournir un équivalent à l'expression*
«boit»
parmi les expressions :
«lève»
usager : *NON*
APLEC : *Pouvez-vous fournir un équivalent sémantique à l'expression :*
«Marie»

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

parmi les expressions :
«enfants» «hommes»
usager : *«enfants»*
APLEC : *Les enfants boivent du lait ?*
Autre question ?
usager : *MERCI*

usager : *Marie boit-elle ?*
APLEC : *Les enfants boivent du lait.*
Autre question ?

usager : *Marie se lève-t-elle tôt ?*
APLEC : *Marie se lève tard.*
Les enfants se lèvent tôt.
Autre question ?

Dans la première intervention, APLEC compare «boit» et «lève», l'action (PRO*) de Marie dans la question et l'action de Marie dans la réponse courante (Marie se lève tard -- seule séquence du texte rapportée par le premier modèle : THE*). Ces deux mots ne sont pas identiques et APLEC demande l'aide de l'usager. Si l'usager à cet endroit avait acquiescé à la proposition d'équivalence, la réponse courante serait devenue réponse finale. Mais l'usager refuse de voir une relation d'équivalence entre les deux lexèmes et indique à APLEC son refus de façon définitive (NOND), de telle sorte qu'APLEC ne le dérangera plus à l'avenir sur cette relation (l'usager aurait pu signaler un refus ad hoc pour cette question (NON plutôt que NOND)).

APLEC change donc l'ordre des modèles ; (THE* PRO*) devient : (PRO* THE*) ; seuls deux modèles sont efficients sur cette question. PRO* ramènera les séquences du texte qui ont «boit» ou «boivent» comme verbe principal du propos (puisque ce verbe est le verbe principal de la question) ; «boit» et «boivent» peuvent être confondus grâce à un algorithme de lemmatisation (basé sur une procédure de troncature par la droite). On tentera maintenant de comparer «Marie» le thème de la question à «enfants» et/ou «hommes» les thèmes de la réponse courante. Ici l'usager propose une équivalence entre «Marie» et «enfants» et APLEC dépistera la réponse finale.

Par la suite, l'exécution de la fonction MERCI projettera la relation d'équivalence proposée sur l'ensemble du texte. APLEC a en quelque sorte appris la relation offerte par l'usager.

Une séance avec APLEC a donc ainsi l'aspect d'une suite de questions posées par l'usager et auxquelles tente de répondre l'apprenti-lecteur. Les questions sont formulées dans la langue du texte. Sur chacune d'entre elles, la GDT de l'usager est appliquée et APLEC essaie par la suite de caractériser au mieux la description obtenue, ceci au moyen des modèles d'exploration préalablement proposés. Les segments du texte qui réalisent ces modèles deviennent les réponses appropriées aux

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

questions posées. Si l'un des modèles soumis par l'usager ne peut servir à fournir directement une réponse, l'apprenti-lecteur cherche quand même à dépister les éléments du texte qui occupent une position analogue à l'élément problématique de la question, puis entre en communication avec l'utilisateur pour s'enquérir auprès de lui d'une solution sémantique au problème dépisté. Nous avons jusqu'à présent expérimenté deux formes générales pour ces solutions, deux formes qui donnent lieu à deux stratégies d'apprentissage différentes. Dans la première APLEC demande à l'usager de confirmer si possible l'existence d'une équivalence sémantique entre l'élément problématique de la question et l'un ou l'autre des éléments dépistés dans le texte. Si l'usager confirme, les segments du texte qui contiennent les équivalents sémantiques dans les positions structurales exigées seront considérés comme appropriés et la suite des tests (modèles) leur sera appliquée.

Dans la seconde forme de solution, APLEC fournit l'élément problématique de la question et les éléments qui occupent des positions similaires dans les séquences dépistées sur le texte, et demande de fournir si possible un réseau sémantique pour chacun de ces éléments. Ces réseaux (appelés EXFAL en Déredec) seront indexés, et la généralisation de ces solutions comme leur apprentissage sera exécutée.

Les solutions peuvent être apprises par APLEC, c'est-à-dire qu'à la demande de l'usager, APLEC redécrit le texte en y généralisant les solutions sémantiques proposées.

L'impression générale qui se dégage de ce système d'analyse de contenu est celle d'un «robot» personnalisé fonctionnant avec les schémas sémantiques (l'ensemble des solutions) de son utilisateur ; schémas qui se construisent au fur et à mesure des conversations avec le texte. Ainsi, contrairement à ces systèmes de questions/réponses où se trouve déterminé à l'avance l'ensemble complet de relations sémantiques admises, et par le fait même le type d'univers textuel sur lequel il sera possible de le projeter, l'apprenti-lecteur permet de définir pour une GDT donnée un système de questions/réponses qui soit à même de construire les seules associations sémantiques dont il aura besoin pour améliorer son travail de fouille.

Il est à souligner de plus que l'étude de la liste de solutions sémantiques renseignera l'usager sur les limites sémantiques d'exploration de sa grammaire. APLEC constitue par là un moyen rapide de connaître ces dernières.

Notre exemple souligne le caractère personnalisé du système : alors que des équivalences sémantiques sont vraisemblables («Marie» = «enfant») pour des usagers donnés dans des contextes d'utilisation donnés, il serait aberrant de les déposer dans un dictionnaire apriorique de la langue.

2.0 Deuxième stratégie d'apprentissage

Dans notre premier exemple, la stratégie d'apprentissage consistait

a) à retenir une relation d'équivalence proposée par l'usager ;

b) à généraliser cette relation d'équivalence pour toutes les expressions atomiques du texte

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

membres de la relation.

De façon plus générale, on peut penser que toute stratégie d'apprentissage comporte les éléments suivants :

a) **identification, analyse et rétention d'une solution** proposée par l'usager au moment de la reconnaissance par le programme d'une difficulté.

b) **généralisation de la solution** proposée et application contextuelle ou pas de la nouvelle solution sur la question et sur tout le texte.

Pour APLEC, toute «solution» consiste à construire des EXFAL. Par exemple, quand «Marie» et «enfants» sont dits être équivalents, deux EXFAL («Marie» (EQUI* («enfants»)) et («enfants» (EQUI* («Marie»))) sont construites puis indexées à la question et à toutes les expressions «Marie» et «enfants» du texte. Par la suite, les modèles de l'usager pourront dépister «enfants» à la place de «Marie» (ou vice-versa) soit dans la poursuite de l'analyse en question, soit dans des explorations de texte relatives à d'autres questions. L'indexation de ces EXFAL sera définitive au moment de l'évaluation de la fonction MERCI. Les EXFAL construites et apprises sont donc composées de deux expressions atomiques, l'une dépistée sur la question, l'autre choisie par l'usager parmi une liste fournie par APLEC.

Nous croyons pouvoir améliorer cette première stratégie d'apprentissage d'au moins deux façons. D'une part, on peut imaginer que l'usager soit intéressé de fournir **lui-même** la ou les expressions qu'il veut voir reliées à l'expression problématique de la question ou aux expressions dépistées par APLEC dans les séquences du texte qui, à ce moment de l'analyse, sont susceptibles d'être des réponses à la question. On peut aussi penser d'autre part que les relations utilisées dans la construction de l'EXFAL soient autre chose que des relations d'équivalence sémantique (par exemple des relations d'inclusion, de prédication, de cause à effet,etc.).

Il s'agit donc d'une stratégie d'apprentissage très différente de celle présentée jusqu'ici.Elle est en quelque sorte plus générale, demande une participation plus élevée de la part de l'usager, mais possède, comme on le verra maintenant un pouvoir inférentiel beaucoup plus grand.

Reprenons le texte décrit par notre grammaire de surface.

Posons la question :

usager : *L'enfant mange-t-il de la soupe ?*
APLEC : *Exfalisez l'expression : «boivent»*
usager : *(«boivent» (CAT* («action (THE* («anime»))*
(COMP1 («comes»)))))*

Ainsi pour l'usager, «boivent» est une action qui prend pour thème quelque chose d'animé et pour objet quelque chose de comestible ...

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

APLEC : *Exfalisez l'expresssion : «lèvent»*
usager : *(«lèvent» (CAT* («action»)))*
APLEC : *Exfalisez l'expression : «mange»*
usager : *(«mange» (CAT* («action» (THE* («anime»))*
(COMP1 («comes»)))))*

A cette étape, APLEC a dépisté les phrases : Les enfants boivent du lait. Les enfants se lèvent tôt. ...comme réponse courante, ceci, à l'aide du modèle THE*.

Note : dans la première stratégie d'apprentissage, APLEC aurait demandé : Pouvez-vous fournir un équivalent sémantique à l'expression «mange» parmi les expressions : «boivent» «lèvent».

On voit bien la faiblesse de la première approche. C'est surtout lorsque l'usager est placé dans l'ignorance totale du texte qu'il explore avec APLEC, qu'il peut se retrouver dans l'hésitation de décider si une équivalence devrait être fournie ou pas. Il se peut fort bien que «mange» et «boivent» soient somme toute des équivalents sémantiques pour ce texte, mais il peut aussi s'avérer que cette identification soit par la suite une cause nouvelle d'ambiguïté (si par exemple le texte porte essentiellement sur des questions de gastronomie ...).

L'idée de base de cette seconde stratégie d'apprentissage est de pouvoir permettre des inférences du type suivant : si l'usager considère que «mange» prend pour thème quelque chose d'animé et pour objet quelque chose de comestible, alors tous les thèmes de «mange» sont des entités animées et tous les objets de «mange» sont des entités comestibles, ceci pour toutes les occurrences de «mange» ; et de surcroît, ces thèmes et ces objets demeurent des entités animées et comestibles dans toutes les phrases du texte, même s'ils ne sont plus thèmes et objets de «mange». La proposition d'une seule EXFAL (au moment de l'interaction avec APLEC) permet ainsi un effet «boule de neige» d'indexations supplémentaires d'autres EXFAL sur le texte.

Poursuivons l'étude de cette deuxième stratégie d'apprentissage à l'aide d'un second exemple ; le traitement grammatical du texte et de la question est toujours celui de notre grammaire de surface.

1) Jean et Marie sont à la librairie.
2) Marie aimerait bien offrir un livre à Jean.
3) Toutefois elle manque d'argent.
4) Marie décide plutôt de glisser le livre sous son manteau.
5) Le Libraire perdra un livre mais Jean sera content.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

usager : *Marie a-t-elle volé ?*
APLEC : *Exfalisez «offrir»*
usager : *(«offrir» (OBJET* («valeur)))*
APLEC : *Exfalisez «glisser»*
usager : *(«glisser» (CAT* («action)))*
APLEC : *Exfalisez «volé»*
usager : *(«volé» (CAT* («action» (OBJET* («valeur»)))))*
APLEC : *Marie décide plutôt de glisser le livre sous son manteau.*
Autre question ?

Au moment où APLEC demande d'exfaliser, le premier modèle THE* a dépisté les phrases du texte qui comme la question, ont «Marie» pour thème (les phrases 1, 2 et 4). Sur cette «réponse courante», APLEC tente de composer le second modèle ; ici, le propos dépisté sur la question, à savoir : «volé», ne se retrouve pas dans la réponse courante, et c'est ce qui provoque les demandes d'exfalisations. Chacune des EXFAL proposées fait l'objet du traitement inférentiel. C'est ainsi que grâce à l'EXFAL («offrir» (OBJET* («Valeur»))), «livre» devient : («livre» (EQUI*(«valeur»))) puisque «livre» est un «objet offert» dans le texte, et ceci, dans toutes les phrases de la réponse courante. Par la suite, le traitement inférentiel sera associé à l'EXFAL proposée sur le mot de la question («volé») ; ce traitement revient à poser sur la réponse courante la demande suivante : existe-t-il une expression «action» ayant un objet (relation P1) avec une expression «valeur». «Glisser» correspond à cette demande puisqu'il a reçu l'équivalence «action» et qu'il prend pour objet l'expression («livre»(EQUI* («valeur»))) dans la phrase 4. «Glisser» et «volé» peuvent ainsi être pris l'un pour l'autre.

Ainsi, pour l'usager : a) on offre habituellement un objet de valeur, «livre» devient par là «objet de valeur», b) glisser est une «action sur un objet de valeur», et c), cette définition correspond à la définition de «voler».

Il est bien entendu que l'exemple ne vise pas à démontrer l'intérêt de cette mini sémantique du «vol». Elle est faible à plusieurs égards : par exemple elle ne suffirait pas à distinguer «voler» de «acheter» puisque les deux mots sont des actions sur des objets de valeur ... Ce qu'il faut souligner, c'est que si «acheter» avait été (dans le texte) une des actions de Marie, APLEC aurait exigé une exfalisation supplémentaire et aurait ainsi obligé l'usager à appronfondir les définitions sémantiques requises. L'exemple met en relief un des principes du fonctionnement d'APLEC : ce n'est qu'au besoin et au fur et à mesure qu'elles sont requises que les définitions sémantiques sont exigées. L'univers des connaissances se construit ainsi petit à petit, en interactif, sans aucun superflu et de manière particulière pour chaque usager, voire pour chaque texte étudié.

Nous ne prétendons pas fournir une théorie achevée de l'apprentissage automatique, mais différentes stratégies formelles fonctionnant quel que soit le contenu des grammaires Déredec expérimentées sur les textes.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

3.0 Le traitement paraphrastique

Dans les exemples présentés jusqu'à maintenant, un même modèle d'exploration permettait d'extraire un lexème de la séquence question (dans une position structurale donnée), puis de dépister les séquences du texte questionné qui contenait ce même lexème dans la même position structurale. Il est possible de modifier cette procédure et de soumettre deux modèles d'exploration différents à APLEC : le premier dépistera un lexème dans une position donnée sur la question, et le second permettra de dépister les séquences du texte qui contiennent ce lexème mais dans une position structurale différente. Ainsi peut-on par exemple demander à APLEC de rassembler comme réponse les séquences d'un texte qui contiennent le thème de la question en position de déterminant nominal dans la réponse.

On peut combiner plus d'un couple de tels modèles d'exploration en fournissant en parallèle à APLEC deux listes de modèles, l'une pour l'analyse de la question, l'autre pour celle de la réponse. Par ce moyen, les relations entre la séquence question et la ou les séquences réponses se trouvent régies par des règles de transformation. Ces règles peuvent s'inscrire à différents niveaux d'intervention grammaticale. Nous y voyons un moyen privilégié pour le dépistage automatique des paraphrases.

Ainsi la transformation paraphrastique dite de nominalisation peut s'exprimer dans les termes de notre grammaire de surface et des deux listes de modèles suivants : (THE* PRO*) et (DETERMINANT* DETERMINE*). Pour qu'il y ait réponse, le thème de la question devra se retrouver en position de déterminant nominal dans la réponse, et le verbe du propos en position de déterminé nominal dans la réponse. Ainsi :

usager : *Jean arrive-t-il ?*

pourra ramener une séquence du type :

APLEC : *L'arrivée de Jean m'étonnait toujours.*

4.0 APLEC v s PLANNER

APLEC associe automatiquement à toute GDT un module d'exploration de type questions/réponses où les questions sont posées dans la langue (naturelle) du texte, et où les réponses sont les segments de celui-ci correspondant le mieux à une série ordonnée de modèles d'exploration fournis par l'usager et représentatifs des régularités indexées par sa GDT, celles du moins dont il veut tester le pouvoir analytique.

Certains systèmes de programmation ayant vu le jour ces dernières années dans les laboratoires d'intelligence artificielle tels PLANNER (Hewitt 1972), MICRO-PLANNER (Sussman 1972), FUZZY (Lefaivre 1977), CONNIVER (Mc-Dermot Sussman 1972), KRL (Bobrow Winograd 1977) partagent avec APLEC l'idée générale de l'application ordonnée d'explorations sur des banques de données. Le

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

pouvoir discriminant des fouilles (la qualité du «pattern-matching») et les structures de contrôle des procédures (récursion, lambdatisation, retour en arrière) sont les deux axes principaux sur lesquels s'appuieraient des comparaisons systématiques entre tous ces langages. Nous ne nous livrerons par ici à un tel examen, mais signalerons toutefois une différence majeure entre APLEC et ces systèmes, qui à l'effet que ces derniers obligent tous que les données d'entrée soient déclarées dans une structure fixe, c'est-à-dire une syntaxe de formation, dont on concèdera qu'elle a une certaine parenté avec la syntaxe de l'anglais (surtout pour le KRL), sans y être pour autant (et selon nous de loin) assimilable, alors que les séquences analysées par APLEC peuvent bien sûr être des suites d'expressions déjà fortement structurées, mais qu'elles peuvent aussi se présenter comme des séquences en langue naturelle.

Dans ce dernier cas, il est très probable que l'usager aura prévu une série d'automates descripteurs des régularités morphologiques, syntaxiques, sémantiques avant l'appel logé à APLEC. Nous voulons insister sur le fait qu'APLEC se compose directement aux descriptions de texte produites par les automates Déredec, et que son utilisation ne nécessite par là la présence d'aucune interface qui transposerait les résultats des analyses morphologiques, syntaxiques etc. des séquences d'entrée en structures déclaratives admissibles aux langages d'exploration. Un bel exemple de synthèse nécessitant de telles interfaces est le Jeu de Blocs de Winograd où un premier logiciel : PROGRAMMAR permet d'effectuer les analyses de base dont certaines régularités sont captées par une interface dite «séman,tique» puis interprétées par MICRO-PLANNER. Ainsi, quand les usagers de ces logiciels (KRL, MICRO-PLANNER ...) veulent construire des systèmes d'**analyse de texte en langue naturelle,** et non seulement des systèmes de **manipulation de concepts,** ils devront d'une part recourir à d'autres logiciels (ATN, PROGRAMMAR ...) autant qu'il en faudra pour programmer toutes les analyses nécessaires au traitement des séquences données en langue naturelle, et d'autre part implanter les interfaces autorisant les passages d'une structure d'information à une autre.

Cet état de chose a d'abord une conséquence pratique : l'accessibilité à ce domaine d'investigation scientifique est retardée pour plusieurs chercheurs puisque la programmation des interfaces oblige une prise de connaissance beaucoup plus étendue des langages de programmation. Il a aussi une conséquence théorique non négligeable : l'absence d'interfaces (comme en Déredec-APLEC) permet un jeu de va-et-vient entre tous les niveaux de description et de d'exploration ; un usager pourrait ainsi par exemple, dans une même séance avec APLEC comparer deux versions d'un système questions/réponses différentes entre elles sur un point aussi rapproché du traitement initial de séquences d'entrée en langue naturelle que disons celui de l'indexation d'une seule catégorie de base (primitif).

Pour nous résumer sur cette comparaison générale, soulignons que le Déredec-APLEC est un système global d'analyse des textes permettant de surcroît la manipulation ordonnée d'explorations, les GDT programmées et les heuristiques de fouille pouvant par la suite être de plus correlées à une logique d'apprentissage ou, si l'on préfère l'expression, à une logique d'inférence automatique.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

(1) Le logiciel Déredec a été développé sur le DEC-SYSTEM-10 de l'Université du Québec à Montréal. Une version fonctionne sur le DEC-SYSTEM-10 de l'Institut Laue-Langevin (centre de recherches en physique nucléaire) de Grenoble. Une adaptation a été implémentée au Centre de Calcul Interuniversitaire de Grenoble sur un Honeywell MULTICS.

(2) Cinq points signent des différences fondamentales entre le Déredec et les autres logiciels dédiés à la programmation des grammaires (nous ne ferons ici que résumer une discussion du «Manuel de l'usager du Logiciel Déredec» (Plante 1981)).
D'abord il est à remarquer qu'une seule structure de rétention de l'information et qu'une seule structure algorithmique de description et de fouille permet un traitement polyvalent où plusieurs descriptions grammaticales inscrites à différents niveaux (morphologie, syntaxe, sémantique, etc.) correspondent sans interfaces.
Deuxièmement, les fonctions qui produisent les descriptions de texte, (c'est-à-dire qui commandent l'exécution des automates sur les séquences), ainsi que les fonctions d'édition qui fouillent les descriptions obtenues (fonctions dites d'exploration) ont entre elles un haut degré de composition; c'est-à-dire que les résultats des fonctions descriptives tout comme les résultats des fonctions d'explorations peuvent à leur tour faire l'objet soit de nouvelles descriptions, soit de nouvelles explorations.
Troisièmement, les automates Déredec privilégient par le jeu des opérations qui y sont programmables l'élaboration de grammaires de type ascendant plutôt que descendant. Nous ne croyons pas les deux démarches simplement inverses l'une à l'autre. Les grammaires descendantes sont, à l'image des grammaires génératives, des algorithmes où l'on tente de rejoindre les items lexicaux des séquences d'entrée par une série de dérivations dont les premières sont très générales et les suivantes de plus en plus particulières. Au contraire, les grammaires ascendantes pensent d'abord des caractérisations particulières des items lexicaux, par exemple des indexations catégorielles ou des regroupements locaux, puis, par un jeu d'applications successives, des caractérisations de plus en plus générales sur les items déjà caractérisés.
Les deux perspectives ne sont pas équivalentes : le programmeur doit nécessairement imaginer dans le cas des grammaires descendantes tous les cheminements qui relieront les caractérisations générales aux items lexicaux, alors que dans ce que nous appelons les grammaires ascendantes, les règles de caractérisation d'un niveau de généralisation donné ne concernent que le comportement ou la distribution des caractérisations du niveau précédent, aucune théorie de l'affiliation dérivationnelle complète n'étant exigée pour qu'une construction descriptive soit quand même amorcée.
Plusieurs directives de programmation des automates Déredec sont basées sur cette question de l'ascendance. Les exigences des grammaires descendantes paraissent beaucoup trop élevées et inappropriées pour l'état actuel de nos connaissance empiriques sur les langues naturelles. Le logiciel permet techniquement de programmer des grammaires descendantes ou même hybrides (descendantes et ascendantes), mais l'usager tirera maximalement profit du système en programmant des automates ascendants.
Quatrièmement, la sensibilité du contexte, qui dans les automates préside l'application des règles, peut être dans les automates Déredec, élargie à l'ensemble du corpus textuel traité. Cela signifie que l'on peut exiger que la description d'un événement textuel tienne compte de l'emplacement particulier de cet événement dans l'ensemble du texte, différenciant de plus la partie du texte qui suit l'événement de celle qui le précède. Associée au pouvoir récursif général de tout le système, cette dimension autorisera l'écriture de grammaires récursives de texte en plus des grammaires récursives phrastiques.
Enfin, cinquièmement et dernier point, certaines caractéristiques formelles de l'ensemble du système rendent possible l'élaboration de procédures de programmation automatique sensibles au contexte. Des fonctions du logiciel permettent en effet de construire automatiquement à partir des résultats de fonctions descriptives ou de fonctions exploratrices, des arguments pour la programmation de nouvelles fouilles ou de nouvelles descriptions.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

(3) La syntaxe BNF des modèles d'exploration est présentée dans PLANTE 1981 (sur le manuel de l'usager du logiciel Déredec). La compréhension formelle de cette écriture n'est pas nécessaire à la lecture de notre présent article. On peut toutefois indiquer que =EA renvoie à «toute expression atomique» c'est-à-dire dans le cas de notre GDT à tout lexème; que le signe – signale que le groupe gauche reçoit la relation de dépendance, et qu'à l'inverse le signe + indique que le groupe gauche origine la relation.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

RÉFÉRENCES

– **BOBROW D.G. and FRASER J.B.** -*An Augmented State Transition Network Analysis Procedure.* in IJCAI-69,1969

– **BOBROW D.G. and WINOGRAD T.** - *An overview of KRL, a Knowledge Representation Language.* in Cognitive Science, 1:1, pp. 3-46, January 1977.

– **BOBROW D.G., WINOGRAD T. and KRL Research Group** - *Experience with KRL-O : One Cycle of a Knowledge Representation Language* in Proceedings of IJCAI-77.

– **CONIN B., COURTINE J.J, GODET F., MARANDIN J.M. et PÊCHEUX M.,** - *Matérialités discursives* (Actes du colloque de Nanterre, Paris X), avril 1980, Lille. PUL, 1981.

– **COURTINE J.J.,** - *Quelques problèmes théoriques et méthodologiques en analyse du discours, à propos du discours communiste adressé aux chrétiens,* Langages, 62, juin 1981 pp. 9-128.

– **DEWAR H., BRATLEY P. and THORNE J.** - *A Program for the Syntactic Analysis of English Sentences* in CACM, vol. 12, n°8, pp. 476-479, 1969.

– **HEWITT C.,** *Description and Theoretical Analysis of PLANNER : a Language for Proving Theorems and Manipulating Models in a Robot.* Ph.D. Thesis, MIT, Camb., Mass., 1971.

– **HEWITT C.,** - *PLANNER : a Language for Manipulating Models and Proving Theorems in a Robot.* IJCAI-69.

LEFAIVRE R., - *FUZZY reference manual.* Rutgers University, March 22, 1977.

LEFAIVRE R., - *RUTGERS/UCI LISP Manual.* Computer Science Department, Rutgers University, April 1978.

– **MEEHAN J.R.,** - *The New UCI LISP Manual.* Lawrence Erlbaum Associates Inc. Publishers; 1979

– **Mc DERMOTT D.V., SUSSMAN G.J.** - *The CONNIVER Reference Manual.* MIT Art. Int. Memo 259a.

– **PÊCHEUX M.,** - *Mises au point et perspectives à propos de l'Analyse automatique du discours.* Langages, 37, mars 1975, pp. 7-80.

– **PLANTE P.,** - *Proposition d'algorithme pour le dépistage de relations de dépendance contextuelle dans un texte.* Thèse de Maîtrise, dépt. de Philosophie, Univ. du Québec à Montréal, août 1975.

– **PLANTE P.** - *Le dépistage automatique des structures de surface et l'analyse des textes philosophiques.* CIRPHO vol. 3, 1975-76, n°2.

– **PLANTE P.** - *Le Système Déredec.* Third International Conference on Computing in the Humanities. Waterloo, 1977.

– **PLANTE P.** - *Le Déredec, manuel de l'usager.* (première édition : 1977, seconde édition : 1979, troisième édition : octobre 1980, quatrième édition : septembre 1981), Service de l'Informatique de l'Univ. du Québec à Montréal.

– **PLANTE P.** - *Un Apprenti-Lecteur.* Joint Conference on Computer Science/Linguistics/Text Processing; Learned Societies; Western Univ. Ontario 78.

– **PLANTE P.** - *Le Déredec, logiciel pour le traitement linguistique et l'analyse de contenu des textes.* 5th International Congress of Applied Linguistics; Montréal 78.

– **PLANTE P.** - *Le Déredec, logiciel pour le traitement linguistique et l'analyse de contenu des textes.* Thèse de Doctorat (Ph.D), Université du Québec à Montréal, Août 1979.

– **PLANTE P.** - *Bibliographie sur l'analyse computationnelle des langues naturelles.* Université du Québec à Montréal; Janvier 1980.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.

– **PLANTE P.** - *Une Grammaire Déredec des structures de surface du français, appliquée à l'analyse de contenu;* Service d'informatique de l'UQAM; Octobre 1980.

– **SHROBE H.E.**- *Dependency Directed Reasoning for Complex Program Understanding.* MIT Report TR-503, April 1979.

– **WINOGRAD T.** - *PROGRAMMAR : a Language for Writing Grammars,* in AI Lab, Memo n° 181, MIT, Camb., Mass., 1969.

– **WINOGRAD T.** - *Understanding Natural Language.* Academic Press, 1972.

– **WOODS W.A.** - *Transition Network Grammars for Natural Language Analysis,* in Communications of the ACM, vol. 13, n° 10, pp. 591-606, Octobre 1970.

– **WOODS W.A.** - *An Experimental Parsing for Transition Network Grammars,* in BBN Report n° 2362, 15 May 1972.

– **ZARRI G.P.** - *Sur le Traitement automatique de données biographiques médiévales : le Projet RESEDA,* in Computing in the Humanities, Univ. of Waterloo. Press, Lusignan S., et North J. (eds), 1977.

– **ZARRI G.P.** - *Building the inference component of an historical information retrieval system.* Proceedings of the seventh international joint conference on artificial intelligence; august 1981, Vancouver Canada, edited by the American Association for Artificial Intelligence, Menlo Park, Calif. 1981.

Actes du Congrès international informatique et sciences humaines
1981 - L.A.S.L.A. - Université de Liège - Tous droits réservés.