

# Génération des anaphores pronominales en dialogue oral homme-machine en Français, utilisant une approche pragmatique-sémantique

Vladimir POPESCU<sup>1,2</sup>, Jean CAELEN<sup>1</sup>, Corneliu BURILEANU<sup>2</sup>

<sup>1</sup> Laboratoire d'Informatique de Grenoble, Institut National Polytechnique de Grenoble, France  
{vladimir.popescu, jean.caelen}@imag.fr

<sup>2</sup> Faculté d'Electronique, Télécommunications et Technologie de l'Information,  
Université "Politehnica" de Bucarest, Roumanie

## 1 Introduction

Le but de cet article est de décrire les démarches d'ordre pragmatique et sémantique à poursuivre lors de la génération des énoncés anaphoriques dans le contexte du dialogue homme-machine. L'idée de générer des anaphores pronominales via des approches motivées par la pragmatique n'est pas nouvelle ; pour des recherches actuelles à cet égard, on peut se rapporter à [20], auxquelles on peut ajouter les recherches de C. Gardent à LORIA et, hors France, des démarches diverses sur ce sujet, utilisant en général des grammaires syntaxiques spécifiées par des règles appropriées à chaque langue d'expression, sont discutées en [16].

La nouveauté des démarches présentées dans cet article réside dans l'appui seulement sur des connaissances pragmatiques (la structure discursive préalablement calculée) et sémantiques (entités dans une ontologie discursive et dans une ontologie de tâche) indépendantes de la langue, et sur des éléments de syntaxe, dépendantes de la langue mais non de l'application. L'approche continue les recherches que notre équipe mène pour l'obtention d'un module de génération s'incluant dans un système de dialogue oral homme-machine. Ainsi, des particularités du dialogue sont prises en compte, telles que la structuration rhétorique spécifique s'établissant dans ce contexte, entre les répliques provenant des usagers différents, partenaires en dialogue.

Pour pouvoir gérer l'emploi des anaphores dans les énoncés textuels on doit descendre dans les spécificités des langues utilisées par le système car les références constituent un phénomène fortement pragmatique, mais à la fois contraint par les spécificités du langage. Ainsi, on essaie de combiner un ensemble de règles gérant des patterns syntaxiques en langue française (et roumaine ensuite) via une grammaire déterministe définie de manière spécifique, avec les spécifications discursives provenant du générateur pragmatique.

Pour cela, on considère que l'élément de départ dans le traitement est le verbe, matérialisant un prédicat dans l'ontologie du domaine. Pour les verbes on va considérer un ensemble de classes, qu'on va définir ci-après. L'essentiel est que ces classes induisent des transformations sur les règles de grammaire, aboutissant à des règles encore plus contraignantes que celles dans la grammaire. Ensuite, en fonction du degré d'"anaphorisation", le patron syntaxique fondamental change ; on le verra en détail comment.

## 2 Règles d'anaphorisation

### 2.1 Grammaire d'anaphorisation

L'objectif de ce paragraphe est de donner un ensemble de règles déterministes faisant le lien entre les entités apparaissant dans le patron syntaxique fondamental et des mots dans la langue, cela tout en passant par les catégories définies dans l'ontologie de domaine et de tâche.

Pour cela, on utilise les notations EBNF ("Extended Backus-Naur Form") standard, avec les particularités suivantes :

- les non-terminaux sont encadrés par des crochets (" $[...]$ ") ;
- les terminaux représentés par des entités dans l'ontologie de domaine et de tâche sont encadrés par des crochets angulaires (" $\langle ... \rangle$ ") ; les relations hiérarchiques entre les entités dans l'ontologie sont représentées avec " $: :$ ", mettant à gauche l'entité hiérarchiquement supérieure et à droite l'entité hiérarchiquement inférieure ;
- le terminal " $\emptyset$ " représente le manque d'instantiation, le manque de valeur, ce qui revient au fait qu'un non-terminal s'expandant vers  $\emptyset$  disparaît ;
- les terminaux représentés par des mots dans la langue sont représentés tels quels, sans être encadrés par des crochets ;
- les non-terminaux du type " $A_{entité}$ " représentent des anaphores pour les entités du type " $entité$ ", où " $entité$ " est " $[agt]$ ", " $[obj]$ ", " $[pat]$ " ou " $[mod]$ ".

Ceci dit, la grammaire d'anaphorisation comprend les règles suivantes :

1.  $[agt] \rightarrow [agt] | [[agt], [CONJ], [agt]] | \emptyset$  ;
2.  $[agt] \rightarrow [agt] | [[det], [agt]] | [A_{agt}]$  ;

3. [det] → [detd][deti] ;
4. [detd] → [detd][A<sub>detd</sub>] ;
5. [CONJ] → et|ou|et aussi ;
6. [detd] → le|la|l'|les ;
7. [deti] → un|une|quelque|certain|certaine|un certain|une certaine|certains|certaines|quelques ;
8. [agt] → ⟨objet⟩ :: ⟨agent⟩ ;
9. [A<sub>detd</sub>] → ce|cette|ces ;
10. [A<sub>agt</sub>] → je|tu|il|elle|nous|vous|ils|elles|celui-ci|celui-là|celle-ci|celle-là|eux-ci|eux-là|celles-ci|celle-là ;
11. obj → [obj] | [[obj], [CONJ], [obj]] | ∅ ;
12. [obj] → [obj] | [[det], [obj]] | [A<sub>obj</sub>] ;
13. [obj] → ⟨objet⟩ :: ⟨chose⟩ ;
14. [A<sub>obj</sub>] → le|la|les|en ;
15. [pat] → [pat] | [[pat], [CONJ], [pat]] | ∅ ;
16. [pat] → [pat] | [[det], [pat]] | [A<sub>pat</sub>] ;
17. [pat] → ⟨objet⟩ :: ⟨agent⟩ | ⟨objet⟩ :: ⟨chose⟩ :: ⟨animé⟩ ;
18. [A<sub>pat</sub>] → me|te|lui|nous|vous|leur ;
19. [mod] → [mod] | [[mod], [CONJ], [mod]] | ∅ ;
20. [mod] → [⟨particule⟩, [obj]] | [mod] | [A<sub>mod</sub>] ;
21. [mod] → ⟨adverbe⟩ | ⟨adverbe⟩, ⟨adverbe⟩ ;
22. [A<sub>mod</sub>] → y|en.

On verra dans les paragraphes suivants le rôle de cette grammaire dans le contrôle de la génération des anaphores.

## 2.2 Patron syntaxique fondamental et son anaphorisation - cas des énoncés non-interrogatifs

Etant donné que les dialogues concernés dans nos recherches sont entre un homme et machine, autour d'une tâche bien définie et comportant une cohérence globale et une pertinence locale, dans le sens de respecter les maximes de Grice [16] on suppose que, pour les énoncés en langue *française* que le générateur puisse produire, le patron syntaxique fondamental est :

[agt] ⟨préd⟩ [obj] ⟨part⟩ [pat] [mod],  
 où "⟨part⟩" représente des mots tels que "à", "vers", "depuis", "aux", cette correspondance étant marquée dans l'ontologie de tâche ; "⟨mod⟩" représente le **modifiant** du prédicat, ce qui correspond en forme linguistique aux adverbes.

Les autres entités intervenant dans l'expression du patron syntaxique fondamental sont :

- le **prédicat** de l'énoncé, noté par "⟨préd⟩" ;
- l'**agent** de l'action indiquée par le prédicat ⟨préd⟩, noté par "[agt]" – l'entité animée réalisant l'action indiquée par le prédicat ;
- l'**objet** de l'action indiquée par le prédicat ⟨préd⟩, noté par "[obj]" – l'entité inanimée faisant l'objet de l'action indiquée par le prédicat ;
- le **patient** de l'action indiquée par le prédicat ⟨préd⟩, noté par "[pat]" – l'entité animée sur laquelle les effets de l'action indiquée par le prédicat s'enregistrent.

Des exemples d'énoncés satisfaisant à ce patron sont (dans le cas où certaines des entités entre crochets peuvent être vides) :

Pierre donne le livre à Jean demain.

Pierre part à Paris demain.

Pierre parle à Marie doucement.

De ces exemples on observe que le patron syntaxique fondamental ne recouvre toutefois pas toutes les situations d'expression, par exemple dans le dernier énoncé ci-dessus, en français on dirait plutôt : "Pierre parle doucement à Marie", ce qui implique une inversion entre le patient et le modifiant, dans la situation où l'objet manque. Au fur et à mesure on pourrait ajouter des patrons syntaxiques fondamentaux et bien évidemment les règles qui s'en suivent.

Selon la situation où une des ou toutes les entités du type agent, objet, patient et / ou précisant sont anaphorisées ou pas le patron syntaxique fondamental se transforme, selon un ensemble de règles qu'on définit dans ce paragraphe. Pour cela on introduit les notations suivantes :

- "∀" – toute valeur possible, autorisée par les règles dans la grammaire d'"anaphorisation" ;
- "∅" – l'entité dont la valeur est ∅ n'existe pas dans la séquence engendrée ;

– “néant” – aucun changement dans le patron syntaxique.

Ceci dit, les règles de transformation du patron syntaxique fondamental, données sous la forme *condition*  $\Rightarrow$  *nouveau patron syntaxique* sont les suivantes :

1.  $\text{equals}([\text{agt}], \forall) \wedge \neg \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \neg \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \neg \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$  néant ;
2.  $\text{equals}([\text{agt}], \forall) \wedge \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \neg \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \neg \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
[agt][obj]<préd><part>[pat][mod] ;
3.  $\text{equals}([\text{agt}], \forall) \wedge \neg \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \neg \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
[agt][pat]<préd>[obj][mod] ;
4.  $\text{equals}([\text{agt}], \forall) \wedge \neg \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \neg \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
[agt][mod]<préd>[obj]<part>[pat] ;
5.  $\text{equals}([\text{agt}], \forall) \wedge \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \neg \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
(a)  $\text{equals}([\text{A}_{\text{pat}}], \text{lui|leur}) \Rightarrow$  [agt][obj][pat]<préd>[mod] ;  
(b)  $\neg \text{equals}([\text{A}_{\text{pat}}], \text{lui|leur}) \Rightarrow$  [agt][pat][obj]<préd>[mod] ;
6.  $\text{equals}([\text{agt}], \forall) \wedge \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \neg \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
[agt][obj][mod]<préd><part>[pat] ;
7.  $\text{equals}([\text{agt}], \forall) \wedge \neg \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
[agt][pat][mod]<préd>[obj] ;
8.  $\text{equals}([\text{agt}], \forall) \wedge \text{equals}([\text{obj}], [\text{A}_{\text{obj}}]) \wedge \text{equals}([\text{pat}], [\text{A}_{\text{pat}}]) \wedge \text{equals}([\text{mod}], [\text{A}_{\text{mod}}]) \Rightarrow$   
(a)  $\text{equals}([\text{A}_{\text{pat}}], \text{lui|leur}) \Rightarrow$  [agt][obj][pat][mod]<préd> ;  
(b)  $\neg \text{equals}([\text{A}_{\text{pat}}], \text{lui|leur}) \Rightarrow$  [agt][pat][obj][mod]<préd>.

Les règles montrent bien comment l’“anaphorisation” induit des transformations sur le patron syntaxique fondamental, dont les éléments constitutifs changent d’ordre, dans la situation où certains d’entre eux sont anaphorisés.

### 2.3 Patrons syntaxiques fondamentaux et leur anaphorisation - cas des énoncés interrogatifs

On pose que les énoncés interrogatifs ont, du point de vue de la machine, pour but de spécifier certains éléments dans un énoncé de l’utilisateur, afin de le rendre complet du point de vue sémantique [5]. Cela revient, en fait, à l’initialisation des variables définies et non-initialisées dans des énoncés provenus de l’utilisateur. Ces variables (à initialiser) peuvent concerner soit l’agent, soit le patient, soit l’objet, soit le modifiant de l’énoncé à anaphoriser. On suppose que le prédicat ne peut être remplacé par une anaphore dans les énoncés interrogatifs (en vertu des maximes de Grice [1], les formules logiques spécifiant l’intention communicationnelle à mettre sous forme linguistique ont au moins le prédicat précisé, initialisé) et que plusieurs des entités agent, patient et modifiant peuvent être anaphorisées dans des énoncés interrogatifs.

Pour aiser la compréhension, on considère d’abord un énoncé non-interrogatif affirmatif complètement spécifié (c’est-à-dire, ayant toutes les entités initialisées) :

Pierre donne le livre à Jeanne demain.

Pour cet énoncé on peut avoir plusieurs types d’énoncés interrogatifs, selon l’ensemble des entités non-initialisées dans des énoncés non-interrogatifs précédents ; ainsi :

1. si l’agent est non-initialisé, l’énoncé interrogatif essayant d’initialiser la variable spécifiant l’agent pourrait être :

Qui donne le livre à Jeanne demain ?

Donc, l’ensemble des règles à appliquer dans la grammaire d’anaphorisation (cf. §2.1), ainsi que les modifications par rapport au patron syntaxique fondamental présenté dans la section antérieure peuvent être précisés dans la règle :

$\text{equals}([\text{agt}], '?') \Rightarrow ([\text{agt}] \rightarrow [\text{A}_{\text{agt}}]) \wedge ([\text{A}_{\text{agt}}] \rightarrow \text{qui}) \wedge ([\text{A}_{\text{agt}}]\langle \text{préd} \rangle [\text{obj}][\text{pat}][\text{mod}]) ;$

2. si l’objet est non-initialisé, l’énoncé interrogatif essayant d’initialiser la variable spécifiant l’objet pourrait être un des énoncés suivants :

Pierre donne quoi à Jeanne demain ?

Qu’est-ce que Pierre donne à Jeanne demain ?

Donc, l’ensemble des règles à appliquer dans la grammaire, ainsi que les deux patrons syntaxiques résultés après la modification du patron syntaxique fondamental peuvent être précisés dans la règle :

$\text{equals}([\text{obj}], '?') \Rightarrow ([\text{obj}] \rightarrow [\text{A}_{\text{obj}}]) \wedge ([\text{A}_{\text{obj}}] \rightarrow \text{quoi}) \wedge (([\text{agt}]\langle \text{préd} \rangle [\text{A}_{\text{obj}}][\text{pat}][\text{mod}]) \vee ([\text{A}_{\text{obj}}]\{\text{est-ce que}\}[\text{agt}]\langle \text{préd} \rangle [\text{pat}][\text{mod}])) ;$

3. si le patient est non-initialisé, l'énoncé interrogatif essayant d'initialiser la variable spécifiant l'objet pourrait être le suivant :  
A qui Pierre donne le livre demain ?  
 Donc, l'ensemble des règles à appliquer dans la grammaire, ainsi que les deux patrons syntaxiques résultés après la modification du patron syntaxique fondamental peuvent être précisés dans la règle :  
 $\text{equals}([\text{pat}], ' ?') \Rightarrow ([\text{pat}] \rightarrow [A_{\text{pat}}]) \wedge ([A_{\text{pat}}] \rightarrow \text{qui}) \wedge (\{\text{\textasciitilde}\}[A_{\text{pat}}][\text{agt}]\langle \text{préd} \rangle [\text{obj}][\text{mod}]) ;$
4. si le modifiant est non-initialisé, l'énoncé interrogatif essayant d'initialiser la variable spécifiant le modifiant pourrait être un adverbe de temps, de mode ou de lieu, ce qui entraîne des patrons syntaxiques divers :
- si le modifiant inconnu concerne le temps de l'action, un énoncé approprié serait :  
Quand est-ce que Pierre donne le livre à Jeanne ?  
 Ceci est obtenu suite à l'application de la règle suivante :  
 $\text{equals}([\text{mod}], ' ?') \wedge \text{equals}(\Delta t(\alpha), ' ?') \Rightarrow ([\text{mod}] \rightarrow [A_{\text{mod}}]) \wedge ([A_{\text{mod}}] \rightarrow \text{quand}) \wedge ([A_{\text{mod}}]\{\text{est-ce que}\}[\text{agt}]\langle \text{préd} \rangle [\text{obj}][\text{pat}]) ;$
  - si le modifiant inconnu concerne le lieu de l'action, l'anaphore sera, à la place de "quand", "où", la règle d'anaphorisation restant la même que ci-dessus ; toutefois, comme on le verra ci-après, la classe du verbe instantiant le prédicat de l'énoncé influe sur le nouveau patron syntaxique ;
  - si le modifiant inconnu concerne la manière dont l'action est effectuée, l'anaphore sera, à la place de "quand", "comment", la règle d'anaphorisation restant la même que ci-dessus ; toutefois, comme on le verra ci-après, la classe du verbe instantiant le prédicat de l'énoncé influe sur le nouveau patron syntaxique.

Pour la situation où plusieurs entités sont non-initialisées dans un énoncé non-interrogatif précédant une interrogation, on suppose que la machine, à chaque question, essaie d'initialiser une seule variable, donc la machine tâche d'initialiser une seule variable à la fois, pour chaque tour de parole (question, dans ce cas). Cela est justifié par le fait que la machine pose des questions seulement afin de compléter les énoncés (incompl $\Delta t$ ) des usagers, ou les plans (incomplets) pour accomplir des tâches. Dans les deux cas, cela revient, en fin de compte, à l'initialisation de variables logiques et, donc, au niveau de chaque question la machine initialise une variable.

### 3 Contraintes induites par la langue : classes de verbes et effets sur l'anaphorisation

A chaque prédicat d'un énoncé (correspondant aussi à un prédicat dans l'ontologie gérée par le contrôleur de tâche) correspond un verbe regroupé dans une classe. Cette classification des verbes se fait (aussi en relation avec les grammaires traditionnelles) par rapport à trois entités faisant partie des énoncés, l'agent, l'objet et le patient (introduits ci-dessus). Ainsi, les huit classes de verbes sont les suivantes :

1. VTD – verbes transitifs directs (il existe l'agent, l'objet et le patient) ; exemple : donner ;
2. VTI 1 – verbes transitifs indirects 1 (il n'existe pas de patient) ; exemple : manger ;
3. VTI 2 – verbes transitifs indirects 2 (il n'existe pas d'objets) ; exemple : masser ;
4. VIT – verbes intransitifs (il n'existe pas d'objet et de patient) ; exemple : dormir ;
5. VI 1 – verbes impersonnels 1 (il n'y a pas d'objet et de patient) ; exemple : pleuvoir ;
6. VI 2 – verbes impersonnels 2 (il n'y a pas de patient) ; exemple faire beau ;
7. VI 3 – verbes impersonnels 3 (il existe l'objet et le patient) ; exemple : falloir ;
8. VI 4 – verbes impersonnels 4 (il n'existe pas d'objet) ; exemple : pleuvoir dans la gorge.

On verra ci-dessous comment l'appartenance du verbe matérialisant le prédicat dans l'ontologie de tâche à une des classes ci-dessus va guider la génération des anaphores.

La classe où le verbe matérialisant le prédicat dans le patron syntaxique fondamental appartient contraint l'ensemble des valeurs que les entités constitutives dans le patron syntaxique fondamental ([agt], [obj], [patient] et [mod]) peuvent prendre. Ainsi, utilisant les mêmes notations qu'auparavant, on peut préciser de manière compacte la relation entre la classe de verbe les valeurs des entités dans le patron syntaxique fondamental, sous la forme d'un ensemble de règles, faites voir ci-dessous :

1.  $\text{equals}(\langle \text{préd} \rangle, \text{VTD}) \Rightarrow \neg \text{equals}([\text{agt}], \emptyset) \wedge \neg \text{equals}([\text{obj}], \emptyset) \wedge \neg \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall) ;$
2.  $\text{equals}(\langle \text{préd} \rangle, \text{VTI 1}) \Rightarrow \neg \text{equals}([\text{agt}], \emptyset) \wedge \neg \text{equals}([\text{obj}], \emptyset) \wedge \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall) ;$
3.  $\text{equals}(\langle \text{préd} \rangle, \text{VTI 2}) \Rightarrow \neg \text{equals}([\text{agt}], \emptyset) \wedge \text{equals}([\text{obj}], \emptyset) \wedge \neg \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall) ;$
4.  $\text{equals}(\langle \text{préd} \rangle, \text{VIT}) \Rightarrow \neg \text{equals}([\text{agt}], \emptyset) \wedge \text{equals}([\text{obj}], \emptyset) \wedge \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall) ;$
5.  $\text{equals}(\langle \text{préd} \rangle, \text{VI 1}) \Rightarrow \text{equals}([\text{agt}], \emptyset) \wedge \text{equals}([\text{obj}], \emptyset) \wedge \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall) ;$
6.  $\text{equals}(\langle \text{préd} \rangle, \text{VI 2}) \Rightarrow \text{equals}([\text{agt}], \emptyset) \wedge \neg \text{equals}([\text{obj}], \emptyset) \wedge \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall) ;$

7.  $\text{equals}(\langle \text{préd} \rangle, \text{VI } 3) \Rightarrow \text{equals}([\text{agt}], \emptyset) \wedge \neg \text{equals}([\text{obj}], \emptyset) \wedge \neg \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall)$  ;
8.  $\text{equals}(\langle \text{préd} \rangle, \text{VI } 4) \Rightarrow \text{equals}([\text{agt}], \emptyset) \wedge \text{equals}([\text{obj}], \emptyset) \wedge \neg \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall)$ .

On observe qu'en fait la classe de verbe *simplifie* le patron syntaxique à anaphoriser (et donc à transformer selon les règles données dans le paragraphe antérieur).

De toutes façons, pour les énoncés interrogatifs, la classe de verbes peut entraîner des modifications sur le patron syntaxique de l'énoncé ; ainsi, dans le cas des verbes intransitifs, une question ciblée à initialiser un modifiant respecte un patron syntaxique spécifié dans la règle ci-dessous :

$$\text{equals}([\text{mod}], ?) \wedge \text{equals}(\langle \text{préd} \rangle, \text{VIT}) \Rightarrow ([\text{mod}] \rightarrow [\text{A}_{\text{mod}}]) \wedge ([\text{A}_{\text{mod}}] \rightarrow \text{quand|où|comment}) \wedge ([\text{agt}]\langle \text{préd} \rangle[\text{A}_{\text{mod}}]).$$

#### 4 Génération des anaphores pronominales

On suppose que le point de départ pour la génération des anaphores pronominales est une structure discursive SDRS (provenant du générateur pragmatique), précisée par les relations rhétoriques comprises, avec les étiquettes et les formes logiques des énoncés constituants. Ainsi, l'obtention d'un texte anaphorique en sortie du générateur suppose le parcours des étapes suivantes :

1. traduire la formule logique de l'énoncé à générer, vers le patron syntaxique fondamental, avec les entités particularisées vers les valeurs des variables dans la formule logique,
2. repérer la classe du verbe instantiant le prédicat dans le patron syntaxique et appliquer la règle de transformation correspondante (cf. §3),
3. parcourir en arrière la SDRS provenant du générateur pragmatique et voir si chaque entité dans le patron syntaxique fondamental avait déjà été introduite dans la SDRS,
4. bloquer l'anaphorisation en cas d'ambiguïté (le référent d'un objet est introduit dans un énoncé antérieur relié par une relation rhétorique à l'énoncé courant, à anaphoriser, mais le référent d'un autre objet, ayant les mêmes catégories grammaticales - genre, nombre, personne, ... existe dans un énoncé situé, dans la SDRS courante, entre l'énoncé précisé ci-dessus et l'énoncé à générer) ;
5. pour chaque entité déjà introduite dans la SDRS, dans un énoncé antérieur à celui qu'on souhaite produire, appliquer la règle d'anaphorisation correspondante (cf. §2.2),
6. générer les anaphores sous forme linguistique, appliquant les règles dans la grammaire d'anaphorisation (cf. §2.1),
7. ajuster la forme linguistique finale pour l'énoncé anaphorisé, s'appuyant sur une grammaire morpho-syntaxique.

L'étape (1) se réalise en principe via l'ensemble des transformations suivantes :

- dans chaque formule logique on repère les paires de prédicats de la forme :  $\text{entité}(\langle \text{variable} \rangle) \wedge \text{equals}(\langle \text{variable} \rangle, \text{value}')$  où *entité* est une des suivantes *agent*, *object*, *patient*, *modifiant* ;
- on effectue les transformations du type :  $\text{entité}(\langle \text{variable} \rangle) \wedge \text{equals}(\langle \text{variable} \rangle, \text{value}') \rightarrow \text{equals}([\text{ent}], \text{value}')$  ;
- en vertu des transformations précédentes, on substitue dans le patron syntaxique fondamental chaque entité [ent] par la valeur *value'* correspondante.

L'étape (2) se réalise selon la discussion dans §3 ; l'étape (3) suppose d'abord l'utilisation d'une grammaire pour représenter les structures discursives SDRS, qu'on précise ci-après. Une telle grammaire suppose que les informations disponibles dans une SDRS provenant du générateur pragmatique concernent : (i) le nombre *D* des énoncés dans la structure discursive *SDRS*, (ii) le nombre *R* des relations rhétoriques reliant les énoncés, (iii) l'ensemble *II* des étiquettes  $\pi$  des énoncés dans la SDRS, (iv) l'ensemble *P* des étiquettes  $\rho$  des relations rhétoriques<sup>1</sup>, (v) les formules logiques *K* des énoncés dans la SDRS, (vi) les sémantiques  $\Sigma$  des relations rhétoriques reliant les énoncés dans la SDRS.

Ainsi, la grammaire de représentation pour les SDRS est précisée par les règles :


1.  $\text{SDRS} \rightarrow (D, R, II, P)$  ;
2.  $II \rightarrow \{\pi_1, \dots, \pi_D\}$  ;
3.  $\pi_i : i \in \{1, \dots, D\} \rightarrow K_i : i \in \{1, \dots, D\}$  ;
4.  $P \rightarrow \{\rho_1, \dots, \rho_R\}$  ;
5.  $\rho_j : j \in \{1, \dots, R\} \rightarrow \rho_j(\pi_i, \pi_k) : j \in \{1, \dots, R\}; i, k \in \{1, \dots, D\} | \rho_j(\pi_i, \rho_k) : i \in \{1, \dots, D\}; j, k \in \{1, \dots, R\}$  ;
6.  $\rho_j : j \in \{1, \dots, R\} \rightarrow \Sigma_j : j \in \{1, \dots, R\}$ .

<sup>1</sup> Les étiquettes des relations rhétoriques reviennent à leurs noms.

Donc, l'étape (3) dans l'algorithme de génération des anaphores pronominales suppose (pour l'énoncé à générer désigné par l'étiquette  $\pi_i$ ) tout simplement le test si l'entité **entité** apparaît, initialisé avec la valeur 'value' dans un énoncé étiqueté  $\pi_j : j < i$  dans la SDRS provenue du générateur pragmatique. Ce test, s'appuyant sur la grammaire des SDRS, se déroule en fait dans deux étapes :


(a) appliquer le test seulement sur les énoncés  $\pi_j$  ayant la propriété qu'une relation rhétorique  $\rho_k$  existe entre eux et  $\pi_i$  et, seulement si le test réalisé à (a) est négatif,

(b) appliquer le test sur tous les énoncés  $\pi_j$  dans la SDRS provenue du générateur pragmatique.

Les étapes (5) et (6) dans l'algorithme de génération des anaphores consistent à appliquer les démarches dans les sections précisées.  Ce qui suit un exemple compréhensif va illustrer toutes ces étapes de traitement.

Jusqu'à présent on n'a pas discuté de l'anaphorisation de l'agent ou de l'anaphorisation du patient, dans le cas où ce dernier est en fait l'allocutaire (le destinataire de l'énoncé). Dans ces deux cas, la recherche des référents appropriés dans la structure SDRS courante est remplacée par des processus inférentiels imposants le choix de certaines règles dans la grammaire d'anaphorisation (cf. §2.1). Ainsi, notant par  $\text{-emitter}(\alpha)$  le destinataire de l'énoncé  $\alpha$ , on a les règles suivantes :

1.  $\text{equals}([\text{agt}], \text{emitter}(\alpha)) \Rightarrow ([\text{agt}] \rightarrow [\text{A}_{\text{agt}}]) \wedge ([\text{A}_{\text{agt}}] \rightarrow \text{je}) ;$
2.  $\text{equals}([\text{agt}], \text{-emitter}(\alpha)) \Rightarrow ([\text{agt}] \rightarrow [\text{A}_{\text{agt}}]) \wedge ([\text{A}_{\text{agt}}] \rightarrow \text{tu}) ;$
3.  $\text{ClassOf}([\text{agt}], \text{emitter}(\alpha)) \wedge \neg \text{Disjoint}([\text{agt}] \setminus \text{emitter}(\alpha), \text{SDRS}_\alpha) \Rightarrow ([\text{agt}] \rightarrow [\text{A}_{\text{agt}}]) \wedge ([\text{A}_{\text{agt}}] \rightarrow \text{nous}) ;$  ici,  $\text{SDRS}_\alpha$  est la SDRS courante, dont  $\alpha$  est le dernier énoncé ;
4.  $\text{ClassOf}([\text{agt}], \text{-emitter}(\alpha)) \wedge \neg \text{Disjoint}([\text{agt}] \setminus \text{-emitter}(\alpha), \text{SDRS}_\alpha) \Rightarrow ([\text{agt}] \rightarrow [\text{A}_{\text{agt}}]) \wedge ([\text{A}_{\text{agt}}] \rightarrow \text{vous}).$

Dans le cas où l'agent ne coïncide ni avec l'émetteur de l'énoncé, ni avec le destinataire, l'algorithme général d'anaphorisation (basé sur la  recherche des référents appropriés dans l'historique du dialogue *relevant, du point de vue rhétorique, pour l'énoncé concerné*).

Pour le patient de l'énoncé dont les anaphores sont à générer on a un ensemble de règles tout-à-fait semblables à celles présentées juste avant, sauf que, dans le cas où le patient coïncide avec l'émetteur de l'énoncé, son anaphorisation revient à l'utilisation du pronom "moi" et, dans le cas où le patient coïncide avec le destinataire, son anaphorisation revient à l'utilisation du pronom "toi".

## 5 Exemple de fonctionnement

Pour le texte d'un énoncé à produire, compte tenu du type de verbe matérialisant son prédicat et du degré de force illocutoire calculé pour cet énoncé (par le générateur pragmatique), on retrouve, dans l'ensemble des règles précisées dans le paragraphe antérieur, la règle correspondante et on l'associe à l'énoncé.

Ensuite, si les entités du type [agt], [obj], [pat] ont été *déjà introduites* dans le dialogue courant<sup>2</sup>, alors appliquer la règle d'anaphorisation correspondante.

Supposons que l'énoncé qu'on doit générer est donné par la formule logique :

$$\exists X, Y, Z : \text{object}(X) \wedge \text{equals}(X, \text{'livre'}) \wedge \text{patient}(Y) \wedge \text{equals}(Y, \text{'Jeanne'}) \wedge \text{good\_time}(Z) \wedge \text{equals}(Z, t_+) \wedge \text{equals}(Z, \text{'demain'})$$

On applique l'étape (1) dans l'algorithme de génération ; cette formule logique permet d'établir les correspondances renfermées dans la proposition logique :

$$\text{equals}([\text{agt}], \emptyset) \wedge \text{equals}(\langle \text{préd} \rangle, \text{need}()) \wedge \text{equals}([\text{obj}], \text{'livre'}) \wedge \text{equals}([\text{pat}], \text{'Jeanne'}) \wedge \text{equals}([\text{mod}], \text{'demain'})$$

Ensuite, au niveau linguistique, s'appuyant sur un dictionnaire, on choisit le verbe français explicitant le prédicat  $\text{need}()$  ; pour cela, on applique une règle embarquée dans la spécification du dictionnaire :

$$\text{equals}(\langle \text{préd} \rangle, \text{need}()) \wedge \text{equals}([\text{agt}], \emptyset) \Rightarrow \text{equals}(\langle \text{préd} \rangle, \text{'il falloir'})$$

Pour déterminer la particule  $\langle \text{part} \rangle$  dans le patron syntaxique fondamental, on applique, au niveau linguistique, la règle :

$$\text{equals}([\text{pat}], \langle \text{objet} \rangle :: \langle \text{chose} \rangle :: \langle \text{nom\_propre} \rangle) \Rightarrow \text{equals}(\langle \text{part} \rangle, \text{'à'})$$

Ensuite, pour déterminer le temps du verbe instantiant le prédicat, on s'appuie à nouveau sur la formule logique en entrée et sur la règle :

$$\text{good\_time}(Z) \wedge \text{equals}(Z, t_+) \Rightarrow \text{equals}(\langle \text{pred} \rangle, \langle \text{temps} \rangle, \text{'futur'})$$

Ainsi, le patron syntaxique fondamental conduit à l'énoncé "Il faudra le livre à Jeanne demain."<sup>3</sup> ; cet énoncé doit maintenant être anaphorisé, suivant l'algorithme pour la génération des anaphores pronominales.

Donc, à l'étape(2) de l'algorithme, étant donné que la classe du verbe "falloir", instantiant le prédicat, est "VI 3", le patron syntaxique subit la contrainte spécifiée sous forme logique ci-dessous :

<sup>2</sup> En fait, on vérifie si les entités syntaxiques ont été déjà introduites dans la structure SDRS courante.

<sup>3</sup> L'article défini pour le mot "livre" est déterminé à partir d'une règle qui spécifie que si l'objet est quantifié par  $\exists$  et sa valeur est spécifiée, alors le mot correspondant est accompagné par l'article défini ; ensuite, s'appuyant sur une grammaire linguistique, on établit que pour le mot "livre" l'article est "le".

$\text{equals}([\text{agt}], \emptyset) \wedge \neg \text{equals}([\text{obj}], \emptyset) \wedge \neg \text{equals}([\text{pat}], \emptyset) \wedge \text{equals}([\text{mod}], \forall)$ .

Lors de l'étape (3) dans l'algorithme de génération, on regarde dans l'ensemble des énoncés antérieurs à l'énoncé qu'on doit générer, dans la même SDRS que celui-ci, si [obj] (dans notre cas le mot "le livre") apparaît de façon explicite, en tant que ⟨objet⟩ :: ⟨chose⟩ ; si oui, alors on autorise la substitution [obj] → [A<sub>obj</sub>], sinon, on ne l'autorise pas. On procède de la même manière pour [pat] (dans notre cas, le mot "Jeanne"). Notant par  $\alpha \in \{0; 1\}$  l'autorisation ( $\alpha = 1$  - substitution autorisée,  $\alpha = 0$  - substitution non-autorisée), on obtient :

$(\exists \alpha : \text{equals}(\alpha, 1) \wedge ([\text{obj}] \rightarrow [A_{\text{obj}}])) \vee (\exists \alpha : \text{equals}(\alpha, 1) \wedge ([\text{pat}] \rightarrow [A_{\text{pat}}]))$ .

La valeur de vérité de chaque proposition, des deux reliées par "∨", est donné par la valeur de  $\alpha$  ; si l'anaphorisation concernée est autorisée par la précédence dans la structure SDRS courante, alors la proposition est vraie. Pour cet exemple on suppose que les deux propositions sont vraies, donc les deux anaphorisations sont autorisées.

Ainsi, lors de l'étape (6) de l'algorithme de génération, on repère dans l'énoncé l'entité (les entités) à substituer par l' (les) anaphore(s) correspondante(s) et on choisit, dans la grammaire d'anaphorisation, les terminaux explicitant les anaphores ; ce processus est piloté par un ensemble de règles dans le dictionnaire (qui correspond en fait à l'ontologie de domaine) :

$([\text{obj}] \rightarrow \langle \text{objet} \rangle :: \langle \text{chose} \rangle) \wedge \text{equals}(\langle \text{objet} \rangle :: \langle \text{chose} \rangle, \langle \text{genre} \rangle, ' \text{masculin}')$

$\wedge \text{equals}(\langle \text{objet} \rangle :: \langle \text{chose} \rangle, \langle \text{nombre} \rangle, ' \text{singulier}')$  ⇒  $([A_{\text{obj}}] \rightarrow \text{le})$  ;

$([\text{pat}] \rightarrow \langle \text{objet} \rangle :: \langle \text{agent} \rangle) \wedge \text{equals}(\langle \text{objet} \rangle :: \langle \text{chose} \rangle, \langle \text{animé} \rangle)$

$\wedge \text{equals}(\langle \text{objet} \rangle :: \langle \text{agent} \rangle, \langle \text{objet} \rangle :: \langle \text{chose} \rangle, \langle \text{animé} \rangle, \langle \text{nombre} \rangle, ' \text{singulier}')$

$\wedge \text{equals}(\langle \text{objet} \rangle :: \langle \text{agent} \rangle, \langle \text{objet} \rangle :: \langle \text{chose} \rangle, \langle \text{animé} \rangle, \langle \text{personne} \rangle, ' 3')$  ⇒  $([A_{\text{pat}}] \rightarrow \text{lui})$ .

Pour l'exemple développé ici, suite à ces traitements on obtient donc, via l'étape (5) :

[agt][obj][pat](préd)[mod], avec :

$([\text{agt}] \rightarrow \emptyset) \wedge ([\text{obj}] \rightarrow [A_{\text{obj}}]) \wedge ([A_{\text{obj}}] \rightarrow \text{le}) \wedge ([\text{pat}] \rightarrow [A_{\text{pat}}]) \wedge ([A_{\text{pat}}] \rightarrow \text{lui})$ .

Cela nous amène à "le lui ⟨préd⟩[mod]", où, effectuant les substitutions pour de chaque entité par la valeur correspondante, calculée lors des étapes précédentes, "Il le lui faudra demain", donc l'énoncé anaphorisé qu'on souhaitait générer.

## 6 Conclusions

Cet article a présenté les démarches pour la génération des textes anaphoriques, dans le cadre d'un générateur des énoncés pour le dialogue homme-machine. L'approche, fortement sémantique, s'appuie toutefois sur des éléments pragmatiques (structure rhétorique du dialogue) et essaie de reléguer autant que possible les traitements purement morfo-syntaxiques. Ainsi, on a spécifié un ensemble des règles dans une grammaire d'anaphorisation et de règles reliant cette grammaire au patron syntaxique fondamental et au type du verbe instantiant le prédicat dans un énoncé. Ensuite, on a proposé et illustré par un exemple étendu un algorithme pour la génération des anaphores pronominales, s'appuyant cependant sur une grammaire de la langue française, dont on ne peut se dispenser complètement. Les démarches décrites ici sont en cours d'implémentation en ISO PROLOG, dans le contexte d'un portail vocal d'entreprise, ayant comme application la réservation des salles de réunion [10].

Dans l'avenir, on devra d'abord étendre le cadre précisé ici afin qu'il inclue les énoncés interrogatifs. Ensuite, le couplage avec le formalisme morfo-syntaxique devra être délimité de manière plus précise et rigoureuse ; on envisage de coupler les démarches décrites ici avec un générateur textuel du français, tel que celui de L. Danlos [9] ou celui de C. Ponton [14], s'appuyant par exemple sur les grammaires LTAG.

## Références

1. ASHER, N., LASCARIDES, A., *Logics of Conversation*, Cambridge University Press, 2003.
2. BERGER, A., *La communication entre agents de communautés mixtes : Un Langage de Conversation Expressif pour agents artificiels*, Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 2006.
3. BILANGE, E., *Dialogue personne-machine*, Ed. Hermès, Paris, France, 1992.
4. CAELEN, J., IMBERDIS, L., "Génération des actes illocutoires pour le dialogue", *Actes du colloque GAT'97*, Grenoble, France, 1997.
5. CAELEN, J., XUEREB, A., *Interaction et pragmatique - jeux de dialogue et de langage*, Ed. Hermès Lavoisier, Paris, 2007.
6. CARBERRY, S., LAMBERT, L., "A Process Model for Recognizing Communicative Acts and Modeling Negotiation Subdialogues", *Computational Linguistics*, vol. 25, no. 1, 1999.
7. CHAMBREUIL, M. (Ed.), *Sémantiques*, Ed. Hermès, Paris, France, 1998.
8. CHU-CARROLL, J., CARBERRY, S., "Collaborative Response Generation in Planning Dialogues", *Computational Linguistics*, vol. 24, no. 3, 1998.
9. DANLOS, L., EL-GHALI, D., "A complete integrated NLG system using AI and NLU tools", *Proceedings of COLING'02*, Taiwan, 2002.

10. FOUQUET, Y., *Modélisation des attentes en dialogue oral*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, 2004.
11. LUGER, G. F., *Artificial Intelligence : Structures and Strategies for Complex Problem Solving*, Addison-Wesley, 2005.
12. MCTEAR, M. F., "Spoken Language Technology : Enabling the Conversational User Interface", *ACM Computing Surveys*, vol. 34, no. 1, 2002.
13. NOGIER, J. -F., *Génération automatique de langage et graphes conceptuels*, Editions Hermès, Paris, 1991.
14. PONTON, C., *Formalisme et architecture d'un système de génération du langage naturel ouvert multi-applications*, Thèse de doctorat de l'Université Stendhal, Grenoble, 1996.
15. POPESCU, V., "Architecture pour la génération du langage naturel en dialogue homme-machine", *Ecole thématique du CNRS "Dialogue et Interaction"*, Autrans, France, 2006.
16. REITER, E., DALE, R., *Building Natural Language Generation Systems*, Cambridge University Press, 2000.
17. RUSSELL, S., NORVIG, P., *Artificial Intelligence : A Modern Approach*, Prentice Hall, 2003.
18. STENT, A., "A Conversation Acts Model for Generating Spoken Dialogue Contributions", *Computer Speech and Language*, 2002.
19. STONE, M., DORAN, C., WEBBER, B., PALMER, M., "Microplanning with Communicative Intentions : The SPUD System", *Computational Intelligence 19*, pp. 311-381, 2003.
20. STRIEGNITZ, K., *Generating Anaphoric expressions : Contextual Reasoning in Sentence Planning*, Ph. D. Thesis, University of Saarlandes, Germany & Université de Nancy, France, 2005.