

## Un modèle de langage pour le DHM : la Grammaire Sémantique Réversible

Jérôme Lehuen

LIUM - Université du Maine  
Avenue Laënnec, 72085 Le Mans Cedex 9  
Jerome.Lehuen@lium.univ-lemans.fr

**Résumé** Cet article propose un modèle de langage dédié au dialogue homme-machine, ainsi que des algorithmes d'analyse et de génération. L'originalité de notre approche est de faire reposer l'analyse et la génération sur les mêmes connaissances, essentiellement sémantiques. Celles-ci sont structurées sous la forme d'une bibliothèque de concepts, et de formes d'usage associées aux concepts. Les algorithmes, quant à eux, sont fondés sur un double principe de correspondance entre des offres et des attentes, et d'un calcul heuristique de score.

**Abstract** In this paper we present a language model for man-machine dialogue, as well as algorithms for analysis and text generation. The originality of our approach is to base analysis and generation on the same knowledge. These one is structured like a library of concepts and syntactic patterns. The algorithms are based on a principle of correspondence between offers and expectations, and calculation of a heuristic scoring.

**Mots-clés :** Grammaire Sémantique, Réversibilité, Analyse, Génération, Dialogue.

**Keywords :** Semantic Grammar, Reversibility, Analysis, Generation, Dialogue.

### 1 Introduction

La finalité de la recherche en dialogue homme-machine (DHM) est de permettre aux systèmes informatiques d'interagir avec leurs utilisateurs selon des modalités proches de celles utilisées dans les interactions humaines, en particulier la modalité verbale. Dans cette optique, les théories et modèles en TAL sont convoqués car ils ont pour vocation de décrire la structure et la fonction des productions langagières. Or, il se trouve que le TAL s'est longtemps focalisé sur l'analyse de textes, selon une approche essentiellement monologique, tant du côté des niveaux de traitements (morphologie, syntaxe, sémantique, pragmatique), que des applications visées (correction, indexation, résumé, traduction). Anne Nicolle écrit dans (Nicolle, 2006) que *la difficulté principale du DHM vient du monologisme de l'interface [...] avec une vision du dialogue comme succession d'énoncés autosuffisants, sur le modèle de l'écrit*. Daniel Luzzati constate dans (Luzzati, 2006) *l'inadéquation des grammaires actuelles construites à partir de données sans rapport avec les productions dialogiques. D'un côté, on trouve des descriptions qui ne correspondent que rarement à des exemples de langue en interaction. De l'autre côté, on a des théories de l'interaction, fortement descendantes, qui ne rencontrent que rarement la parole*. De plus, la morphosyntaxe du dialogue est approximative. Même le concept de phrase est remis en question : *en dialogue, on produit des énoncés dont la complétude ne peut se me-*

*surer qu'en termes d'efficacité interactive* (Luzzati, 2006). Ils comportent des bruits, des hésitations, des *disfluences*, des inattendus structurels. C'est une langue par essence difficile, voire impossible à décrire de façon exhaustive, qui se définit par sa spontanéité et par sa variabilité. Par ailleurs, les systèmes de DHM dissocient souvent les aspects interprétation et génération. Cette dichotomie n'est pas réhibitoire : la génération des énoncés de la machine est moins problématique que l'analyse des énoncés de l'utilisateur, puisqu'on peut déterminer un champ lexical et des formes d'usage en fonction d'une application visée. Cela dit, comme à d'autres auteurs (Gertjan, 1993) (Dymetman, 1994), il nous est apparu intéressant de mutualiser les connaissances de l'analyse et de la génération, le système étant en mesure de dire ce qu'il comprend et de comprendre ce qu'il dit. Cette capacité n'est pas neutre. Tout d'abord, elle permet des citations et reformulations des énoncés de l'utilisateur par la machine. Ensuite, elle rend possible une forme de réflexivité de l'activité langagière de la machine, ce qui lui permet potentiellement de s'autocontrôler au cours des phases de génération. Enfin, elle permet d'envisager des évolutions conjointes de ces deux compétences dès lors que les connaissances évoluent, par intervention extérieure ou par apprentissage automatique.

Partant de ces considérations, nous développons actuellement le moteur de dialogue YADE, construit autour d'un modèle de langage adapté à la problématique du DHM. Ce modèle « orienté connaissances » associe une sémantique lexicale et des bibliothèques de formes d'usage utilisables en analyse comme en génération. Les algorithmes d'analyse et de génération fonctionnent de façon opportuniste. Ils ne sont pas fondés sur des principes d'unification et de contraintes mais sur des principes de dépendances et de score. L'utilisation conjointe d'une syntaxe partielle/relâchée et d'une sémantique lexicale a déjà été explorée, notamment par (Goulian, Antoine, 2001). La spécificité de notre approche tient essentiellement dans la *réversibilité*, laquelle conditionne le modèle des connaissances et les algorithmes. La mise en œuvre d'une grammaire réversible en génération est également explorée dans (Gardent, Kow, 2007) : le générateur GENI combine un algorithme non-déterministe et un mécanisme de sélection des paraphrases. La différence avec notre proposition réside dans le modèle de langage et dans les critères de *déterminisation*. GENI exploite une grammaire d'arbres adjoints, basée sur l'unification, alors que YADE utilise une grammaire essentiellement sémantique.

Le moteur de dialogue YADE est architecturé selon quatre niveaux (cf. Fig. 1) dont les trois premiers (langage, dialogue et tâche interactive) communiquent via une mémoire de travail commune, un peu à la façon des architectures de type « tableau noir ». Cet article se focalise sur le niveau langagier qui implémente la Grammaire Sémantique Réversible (GSR). La première partie décrit le modèle de langage (représentations et connaissances), la seconde partie présente les algorithmes d'analyse et de génération, ainsi que quelques exemples.

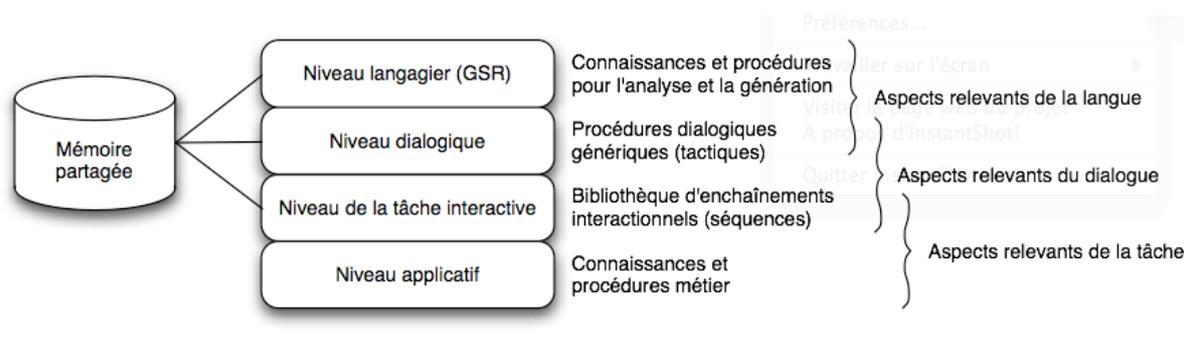


Figure 1 : Architecture du moteur de dialogue YADE

## 2 Le modèle de langage de la GSR

Cette partie présente d’abord le modèle de représentation des énoncés (de l’usager et de la machine), puis le formalisme « orienté connaissances » qui permet de passer d’une chaîne de caractères à une représentation exploitable par le moteur de dialogue et inversement. Ces représentations sont centrales dans notre proposition car elles sont communes aux modules d’analyse, de génération, et de dialogue.

### 2.1 Représentation des énoncés

En DHM, il s’agit d’obtenir, à partir d’énoncés relativement courts, des représentations qui permettent de déclencher des mécanismes cognitifs, en rapport avec une tâche donnée ou avec le dialogue lui-même. À moins d’être une étape nécessaire, une représentation syntagmatique n’est d’aucune utilité. Enfin, nous souhaitons pouvoir passer d’un énoncé à sa représentation et inversement sans avoir à convoquer des modélisations morphosyntaxiques subtiles. Partant de ces objectifs et contraintes, nous avons opté pour un formalisme fortement inspiré des grammaires de dépendance (Tesnière, 1959) mais résolument orienté vers la sémantique. La représentation d’un énoncé consiste en un graphe de relations qui couvre les mots ou expressions significatives par rapport au contexte applicatif et/ou dialogique. Les nœuds du graphe (appelés granules) sont des concepts instanciés dans le double contexte de l’énoncé et du dialogue. Les granules sont identifiés par leur concept, une catégorie sémantique ou pragmatique indiquant son rôle dans la structure ou dans l’interaction, et un numéro d’instanciation.

Par exemple, l’analyse de l’énoncé « *Bonjour, je voudrais une baguette bien cuite s’il vous plaît* » produit trois granules-racines : ouverture:SALUER#1, demande:DEMANDER#6 et politesse:SVP#4 (cf. Fig. 2). Le choix des concepts et des catégories ne dépend pas du modèle : il relève d’un travail d’ingénierie des connaissances à partir d’un domaine d’application et éventuellement d’un corpus de dialogue.

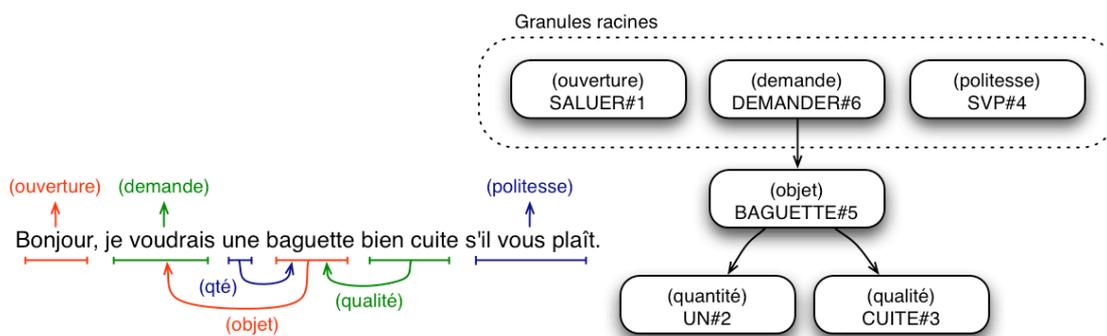


Figure 2 : Structure de dépendances produite par la GSR

### 2.2 Représentation des connaissances

Le modèle des connaissances est construit autour de la notion de *concept*. Un concept représente un objet, une action, la valeur d’un attribut, *etc.* Il possède des propriétés qui permettent, d’une part de le distinguer des autres concepts, d’autre part de le relier à d’autres concepts dans le cadre de l’application visée. Un concept est caractérisé par des offres, des attentes, et des formes d’usage (cf. Fig. 3).

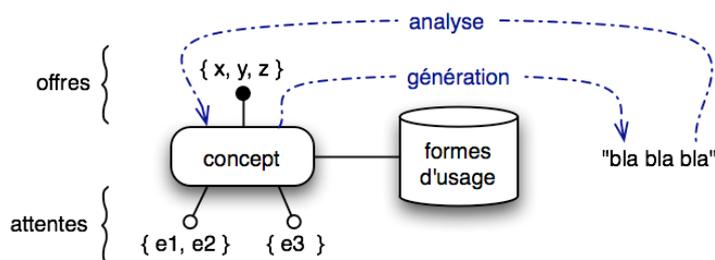


Figure 3 : Modèle de concept de valence 2

Les descripteurs des *offres* sont des catégories ou des traits sémantiques (Katz, Fodor, 1963). Le modèle ne présuppose pas la façon dont les traits sont déterminés : soit à partir de conditions nécessaires et suffisantes (et nous obtenons une collection de *concepts autosuffisants*), soit à partir de différences entre concepts (et nous obtenons une *structure relationnelle de concepts*). Cette distinction, explorée dans (Nicolle et al., 2001), ne fait pas l'objet de cet article. Les *attentes*, qui sont décrites dans les mêmes termes que les offres, caractérisent des actants, des circonstants, des modificateurs, des attributs. Les attentes ne sont pas obligatoires, elles participent à la description du concept en décrivant les liaisons potentielles entre granules. Enfin, un concept est associé à un catalogue de *formes d'usage*. Les formes d'usage sont des structures de surface composées de mots et de références aux attentes du concept. Elles sont associées à des traits pragmatiques et/ou morphologiques qui permettent de caractériser finement les formulations de l'usager pour un même acte de langage (type de langage, politesse, force illocutoire, etc.). La multiplication des formes d'usage engendre une certaine variabilité en interprétation ou pour la génération des énoncés de la machine. Nous avons choisi d'intégrer les variations morphologiques (genre et nombre) dans les formes (codage *full-form*) pour faire l'économie de traitements morphologiques en analyse, et surtout en génération. C'est la raison pour laquelle des traits morphologiques sont combinés aux traits pragmatiques.

```

<concept nom="DEMANDER" offres="action demande">

  <attente code="A1" catégories="demandable"
    description="l'objet de la demande"
    question="que désirez-vous ?"/>

  <syntaxe traits="familier">file-moi A1</syntaxe>
  <syntaxe traits="normal">je voudrais A1</syntaxe>
  <syntaxe traits="normal">avez-vous A1</syntaxe>
  <syntaxe traits="soutenu">puis-je avoir A1</syntaxe>
  <syntaxe traits="assertion rpv">vous désirez A1</syntaxe>
  <syntaxe traits="vérification rpv">c'est bien A1 que vous désirez ?</syntaxe>
</concept>

<concept nom="BAGUETTE" offres="objet demandable">

  <attente code="A1" catégories="quantité"
    description="la quantité"
    question="combien de baguettes désirez-vous ?"/>
  <attente code="A2" catégories="cuisson"
    description="la cuisson"
    question="quelle cuisson souhaitez-vous ?"/>

  <syntaxe traits="féminin singulier">A1 baguette A2</syntaxe>
  <syntaxe traits="féminin pluriel">A1 baguettes A2</syntaxe>
</concept>

```

Tableau 1 : Fragment d'une base de connaissances codée en XML

La constitution d'une base de connaissances dédiée représente un important travail, même s'il est envisageable de réutiliser des « connaissances générales » d'une application à l'autre, que ce soit au niveau de l'ontologie (contribution de l'ingénierie des connaissances) ou au niveau des formes d'usage (contribution de la linguistique). Parallèlement à notre démarche, et non sans rapports avec celle-ci, a été développée la Grammaire Interactive (cf. § 2.3). Cette dernière peut constituer une réponse au problème de la systématisation de la constitution des formes d'usage de la GSR, avec un point de vue linguistique et dialogique.

### 2.3 GSR et Grammaire Interactive

L'objectif de la Grammaire Interactive (Luzzati, 2007) est d'aborder la morphosyntaxe avec un point de vue résolument dialogique : *la morphosyntaxe d'une question se justifie par l'existence d'une réponse présupposée ou escomptée, à laquelle les interactants adhèrent ou dont ils se démarquent*. Le principe consiste à faire varier une formulation prototypique selon des paradigmes qui dépendent ou non du type et du thème de l'énoncé. La combinatoire des critères identifiés génère une table à n dimensions qu'il s'agit de remplir. Par exemple, pour les questions quantificatrices, on fait varier la formulation « *Combien coûte X ?* » selon deux dimensions : une dimension propre à la quantification, et une dimension généralisable à l'interrogation (cf. Tableau 2). Cette étude s'est focalisée pour l'instant sur les questions quantificatrices et locatives du corpus Ritel (Rosset, Petel, 2006).

question	adverbiale	numérale	déterminative	nominale	propositionnelle
<b>simple</b>	<i>Combien coûte X ?</i>	<i>Combien d'euros coûte X ?</i>	<i>Quel prix coûte X ?</i>	<i>Que coûte X ?</i>	<i>X coûte-t-il 5 euros ?</i>
<b>tonique</b>	<i>X coûte combien ?</i>	<i>X coûte combien d'euros ?</i>	<i>X coûte quel prix ?</i>	<i>X coûte quoi ?</i>	
<b>renforcée</b>	<i>Combien est-ce que coûte X ?</i>	<i>Combien d'euros est-ce que coûte X ?</i>	<i>Quel prix est-ce que coûte X ?</i>	<i>Qu'est-ce que coûte X ?</i>	<i>Est-ce que X coûte 5 euros ?</i>
<b>périphrastique</b>	<i>Je voudrais savoir combien X coûte</i>	<i>Je voudrais savoir combien d'euros coûte X</i>	<i>Je voudrais savoir quel prix coûte X</i>	<i>Je voudrais savoir qu'est-ce que X coûte</i>	<i>Je voudrais savoir si X coûte 5 euros</i>
<b>assertive</b>	<i>X coûte 5 euros ?</i>	<i>X coûte 5 en euros ?</i>	<i>Le prix de X est de 5 euros ?</i>	<i>X coûte quelque chose ?</i>	

Tableau 2 : Variabilité de la question « *Combien coûte X ?* » (Luzzati, 2007)

Pour élargir la couverture morphosyntaxique, il suffit d'ajouter des critères de variabilité. Par exemple, la prise en compte de l'adverbe « *bien* » augmente la combinatoire. Après avoir fait le même travail pour les réponses associées, il s'agit de dégager les structures syntaxiques, et de les coder en tant que formes d'usage.

La Grammaire Interactive repose sur l'identification de structures de surface, à partir d'une combinaison de critères morphosyntaxiques et pragmatiques. Elle est moins générique que des grammaires qui reposent sur des connaissances de granularité plus fine. Ceci dit, nous formulons l'hypothèse que la subtilité linguistique permise par la GI, la finesse dans la formulation des réponses, peuvent compenser ce manque de généralité. C'est la raison pour laquelle nous nous inspirons de la GI pour développer la GSR.

### 3 Analyse et génération des énoncés

L'adéquation des représentations proposées au §2 pour le dialogue dépend des processus qui vont les utiliser. En situation de dialogue, en compréhension comme en génération, un résultat erroné ou approximatif est préférable à une absence de résultat qui entraverait l'interaction. Ceci est vrai à condition de pouvoir corriger ou compléter ce résultat. C'est pour cette raison que : premièrement, nous avons privilégié des algorithmes non-déterministes, deuxièmement, les représentations produites par l'analyseur sont partagées avec le dialogueur et manipulables par celui-ci. Les sous-sections suivantes décrivent les algorithmes d'analyse des énoncés de l'utilisateur, puis de génération de ceux de la machine.

#### 3.1 Algorithme d'analyse

L'analyse part d'un énoncé sous forme de chaîne de caractères et produit une représentation construite suivant le modèle présenté au §2.1. L'algorithme d'analyse procède en cinq étapes au cours desquelles il s'agit d'instancier et de relier des « granules » en se fondant sur un critère syntaxique (reconnaissance de leurs formes d'usage) et/ou sur un critère sémantique (correspondance entre leurs offres et leurs attentes). Ces deux critères sont soit combinés, soit utilisés indépendamment l'un de l'autre, afin de générer des hypothèses. Les cinq étapes de l'analyse sont les suivantes :

1. **Noyautage** = instanciation de certains types de granules terminaux (comme les dates) à l'aide d'expressions régulières ou de grammaires locales ;
2. **Segmentation** = identification des portions de texte susceptibles de contenir un granule-fils à l'aide des formes d'usage et des attentes. L'objectif est de générer des hypothèses sur des mots ou expressions non répertoriées (granules hypothétiques) ;
3. **Construction** = instanciation des granules et de leurs liaisons à l'aide des formes d'usage et des attentes. L'algorithme repose sur le remplissage d'un tableau de granules (lexicalisés ou hypothétiques), et sur une fonction d'évaluation ;
4. **Résolution** = suppression des conflits de position entre granules. Les granules faibles sont supprimés au profit des granules forts (cf. fonction d'évaluation) ;
5. **Sauvetage** = rattachement hypothétique des « granules orphelins » sur des critères de correspondance entre offres et attentes, et de proximité.

La méthode choisie pour progresser dans l'analyse sans avoir à effectuer d'incessants retours arrières est l'analyse tabulaire ou *chart parsing* (Kay, 1980) dont le principe est de poursuivre en parallèle des analyses concurrentes, à chaque niveau de granularité permis par le modèle de langage. Dans notre cas, il s'agit de mémoriser les granules (hypothétiques ou non) dans une sorte de tableau-agenda avec leur position, leurs dépendances, et leur score. À la fin de la cinquième étape, ne restent que les structures fortes ou de même poids. En cas d'ambiguïté, une reformulation est proposée et c'est le dialogue qui devra permettre de trancher.

Un granule est caractérisé par une position, une couverture (nombre de mots), et une dispersion (nombre de mots non pris en compte). De plus, il est éventuellement relié à des granules-fils  $G_i$ . Le score d'une liaison  $A_i$  correspond au nombre d'éléments en commun entre les ensembles de traits offerts et attendus. Le score d'un granule  $G$  est donné par la formule :

$$score(G) = couverture(G) - dispersion(G) + \sum_{i=1}^{val} (10 \times score(A_i) \times score(G_i))$$

Par exemple, le score du granule BAGUETTE de l'énoncé : « *Je voudrais une baguette très bien cuite* » est de  $5-1+10+20=34$  (cf. Fig. 4). Tout granule en conflit de position avec celui-ci, et ayant un score inférieur à 34, sera supprimé du *chart*, avec des suppressions en chaîne sur les liaisons et granules dépendants.

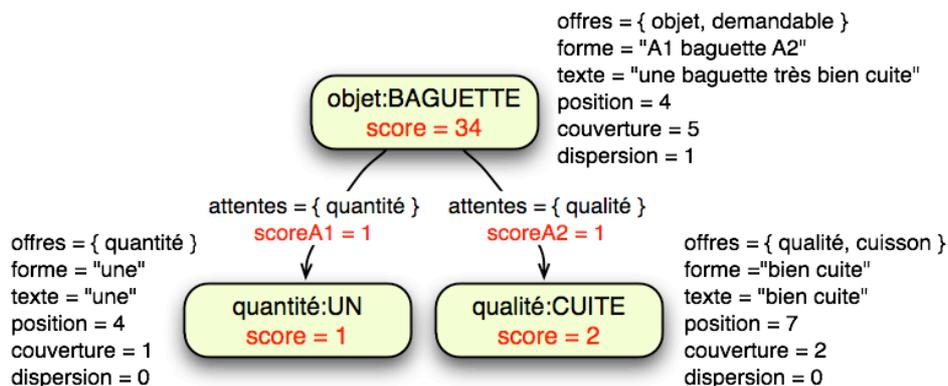


Figure 4 : Calcul du score du granule BAGUETTE

### 3.2 Quelques cas particuliers d'analyse

Cette section donne des exemples qui illustrent des performances particulières (identification des questions toniques/périphrastiques, résolution d'ambiguïtés homonymiques, résolution de conjonctions de coordination), ainsi que la capacité qu'à l'analyse de générer des hypothèses (hypothèses de rattachement ou hypothèses lexicales).

- Le premier exemple illustre la différence d'analyse entre une question tonique et une question périphrastique. Le type des questions non-périphrastiques est noté comme traits d'un granule, alors que les questions périphrastiques génèrent un concept supplémentaire :

a) *Le s2 il commence quand ?*

=> (DATE-DE **interrogation tonique** A1:(événement:S2))

b) *Je voudrais savoir à quelle date commence le s2*

=> (**QUESTIONNER** A1:(information:DATE-DE A1:(événement:S2)))

- Le deuxième exemple montre comment l'analyseur résout naturellement les rares (et artificiels) cas d'ambiguïté homonymiques si le contexte le lui permet. Ici, le vocable « avocat » est désambiguïsé par son adjectif :

c) *Je voudrais un avocat bien mûr*

=> (DEMANDER A1:(demandable:**AVOCAT-FRUIT** A1:(quantité:UN) A2:(avancement:MUR)))

d) *Je voudrais un avocat compétent*

=> (DEMANDER A1:(demandable:**AVOCAT-PERS** A1:(quantité:UN) A2:(qualificatif:COMPETENT)))

- La conjonction de coordination « et » est construite en fonction des catégories ou des traits des granules situés à gauche et à droite de la conjonction, qui déterminent sa portée. La catégorie du concept ET est donnée par ses arguments :

e) *Je voudrais une baguette et avez-vous deux croissants ?*

=> (**demande:ET** A1:(demande:DEMANDER A1:(demandable:BAGUETTE A1:(quantité:UN)))  
A2:(demande:DEMANDER A1:(demandable:CROISSANT A1:(quantité:DEUX))))

f) *Je voudrais une baguette et deux croissants*

=> (DEMANDER A1:(**demandable:ET** A1:(demandable:BAGUETTE A1:(quantité:UN))  
A2:(demandable:CROISSANT A1:(quantité:DEUX))))

- L'étape de sauvetage permet d'analyser des énoncés disloqués qui produisent des granules orphelins. Lorsque les offres et les attentes sémantiques le permettent, l'analyseur tente des rattachements hypothétiques (indiqués ci-dessous par un point d'interrogation) :

g) *Le semestre 2 je voudrais savoir la date*

=> (QUESTIONNER A1:(information:DATE-DE **?A1:**(événement:S2)))

h) *Le semestre 2 la date je voudrais savoir*

=> (QUESTIONNER **?A1:**(information:DATE-DE **?A1:**(événement:S2)))

- Enfin, grâce à l'étape de segmentation, l'analyseur est capable de faire des hypothèses sur des mots inconnus. Les segments [le truc] et [le bidule] ont été générés à partir de la forme d'usage « ranger [objet] dans [rangement] ». Au dialogueur de gérer ces hypothèses :

i) *Ranger le truc dans le bidule*

=> (RANGER A1:(**?objet:"le truc"**) A2:(**?rangement:"le bidule"**))

### 3.3 Algorithme de génération

Le principe est de reconstruire un texte à partir d'une structure de granules en activant les formes d'usage les plus intéressantes et en les composant. Comme pour l'analyse, il s'agit d'être opportuniste, c'est-à-dire de faire pour le mieux avec ce dont on dispose, en fonction d'indices et non pas de contraintes. L'algorithme de génération procède de la façon suivante. Un but de génération est propagé de la racine de la structure vers ses feuilles, en véhiculant un ensemble  $Tg$  de modificateurs globaux (cf. Fig. 5). Un score est calculé, pour chaque verbalisation et pour chaque nœud, en fonction d'un critère global (adéquation avec  $Tg$ ), et d'un critère local (score des fils et nombre de traits communs entre pères et fils). L'énoncé final est obtenu en combinant les verbalisations de meilleurs scores. Soit  $cote(V)$  la cote d'une verbalisation,  $Tp(V)$  ses traits propres, et  $Tp(Vi)$  les traits propres des verbalisations « filles », le score d'une verbalisation  $V$  est alors donné par la formule :

$$score(V) = cote(V) + card(Tp(V) \cap Tglob) + \sum_{i=1}^{val} (10 \times score(Vi) + card(Tp(Vi) \cap Tp(V)))$$

Par exemple, le score de la verbalisation « *une baguette bien cuite* » est de  $1+0+12+12=25$ , tandis que le score de la verbalisation « *un baguette bien cuit* » est de  $1+0+11+11=23$ . Avec  $Tg = \{\text{familier}\}$ , le score de la verbalisation « *je voudrais une baguette bien cuite* » est de  $1+0+250=251$ , tandis que le score de la verbalisation « *file-moi une baguette bien cuite* » est de  $1+1+250=252$ . Cette verbalisation est gagnante. Si l'expression « *file-moi...* » n'avait pas été connue du système, « *je voudrais...* » serait restée. En cas de ballottage, on applique une

fonction aléatoire. La cote d'une verbalisation (par défaut = 1) permet de faire évoluer les « habitudes » du générateur en fonction de critères extérieurs au processus de génération, comme la fréquence d'usage constatée en analyse, par exemple.

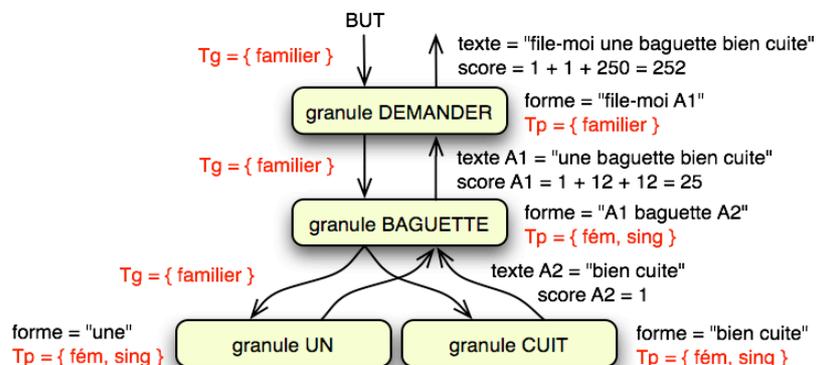


Figure 5 : Exemple de processus de génération

Les algorithmes d'analyse et de génération ont été testés sur des jeux d'énoncés, à l'aide d'un script de « double traduction », chaque énoncé étant analysé puis régénéré dans la foulée. La représentation interne de l'énoncé est alors utilisée comme un « langage pivot ». Nous avons ainsi pu comparer les énoncés d'origines et leurs régénérations. Bien que notre problématique n'ait *a priori* aucun rapport avec la traduction automatique, nous avons tenté de fournir au système des formes d'usage anglophones et spécifié un  $Tg = \{\text{anglais}\}$ . Il a été intéressant de constater que le générateur produisait des traductions correctes et que, lorsqu'une forme d'usage anglophone venait à manquer, l'algorithme utilisait une forme d'usage francophone. Le mélange d'anglais et de français qui en résulte illustre bien le principe opportuniste.

## 4 Conclusion

La finalité de notre modèle n'est pas de représenter la phrase ou l'énoncé. Elle est de fournir à un moteur de dialogue finalisé les éléments qui lui permettront de répondre à un énoncé, dans le contexte d'une application donnée. Ainsi, les représentations produites sont ni absolues, ni forcément utilisables hors d'un contexte dialogique. De plus, elles sont très dépendantes des connaissances fournies, lesquelles sont très dépendantes de l'application visée. En revanche, nous proposons un formalisme « réversible » puisque utilisable en analyse comme en génération, et des algorithmes indépendants des applications. Parce qu'il tolère bruits, dislocations et incomplétudes, et parce qu'il peut faire certaines hypothèses, l'analyseur vise la robustesse. Le modèle de langage est non-normatif car les structures syntaxiques décrites sont davantage proposées qu'imposées. Nous l'avons également souhaité extensible : il suffit d'ajouter une forme d'usage pour augmenter son potentiel, en analyse comme en génération. La multiplication des formes d'usage permet d'interagir en prenant en compte la variabilité et les subtilités langagières décrites dans la grammaire interactive. Le modèle de langage et les algorithmes présentés sont implémentés en CLIPS (<http://clipsrules.sourceforge.net>) et opérationnels. Nous disposons également d'un éditeur dédié permettant d'éditer une ontologie de concepts selon le modèle ANADIA (Beust, 1998) (Nicolle et al. 2001), ainsi que sa GSR associée. La plate-forme de dialogue est en cours de développement. Elle intègrera un module de reconnaissance de la parole et un module de synthèse vocale. Afin de positionner notre approche par rapport à d'autres, nous avons engagé une évaluation en nous fondant sur le corpus utilisé pour la campagne Evalda-Media (Bonneau-Maynard et al. 2006).

## Références

- BEUST P. (1998). *Contribution à un modèle interactionniste du sens*. Thèse de doctorat d'informatique de l'Université de Caen.
- BONNEAU-MAYNARD, H., AYACHE, C., BECHET, F., DENIS, A., KUHN, A., LEFEVRE, F., MOSTEFA, D., QUIGNARD, M., ROSSET, S., SEVRAN, C., VILLANEAU, J. (2006). Results of the french evalda-media evaluation campaign for literal understanding. In: *Proceedings of LREC 2006*, Genoa, 2054–2059.
- DYMETMAN M. (1994). Inherently reversible grammars. In: Tomek Strzalkowski (ed.), *Reversible Grammar in Natural Language Processing*, Kluwer, 33–57.
- GARDENT C., KOW E. (2007). A symbolic approach to near-deterministic surface realisation using tree adjoining grammar. In: *Proceeding of ACL 2007*, Prague, 328–335.
- GERTJAN VON NOORD (1993). *Reversibility in Natural Language Processing*. Dissertation. University of Utrecht. <http://www.jet.rug.nl/~vannoord/papers/diss/diss.html>
- GOULIAN J., ANTOINE J.-Y. (2001). Compréhension Automatique de la Parole combinant syntaxe locale et sémantique globale pour une CHM portant sur des tâches relativement complexes., In: *Proceeding of TALN 2001*, Tours, 203–212.
- GREIMAS A.J. (1966). *Sémantique Structurale*. PUF : Paris.
- KATZ J.J., FODOR J.A. (1963). The Structure of a Semantic Theory. In: Rosenberg and Travis (1971), *Language* n°34 (2). 170–210.
- KAY M. (1980). *Algorithm shemata and data structures in syntactic processing*. Rapport CSL-80-12. Xerox Corporation.
- LUZZATI D. (2006). Dialogue et apprentissage. In: *Compréhension des langues et interaction*, Gérard Sahab (ed.), Hermès-Lavoisier, 337–356.
- LUZZATI D. (2007). Essai de description interactive des questions quantificatrices. In: Actes du colloque *La Quantification*. Strasbourg. Presses Universitaires de Caen. (à paraître)
- NICOLLE P., BEUST P., PERLERIN V. (2001). Un analogue de la mémoire pour un agent logiciel interactif. In: Revue *In Cognito* n°21. 37–66.
- NICOLLE P. (2006). Compréhension et interaction. In: *Compréhension des langues et interaction*, Gérard Sahab (ed.), Hermès-Lavoisier, 141–170.
- ROSSET S., PETEL. S. (2006). The Ritel Corpus – An annotated Human-Machine open-domain question answering spoken dialog corpus. In: *Proceedings of International Conference on Language Resources and Evaluation*. Gênes. 1640–1643.
- TESNIERE L. (1959). *Éléments de syntaxe structurale*, Paris, Klincksiek.