

Contrôle rhétorique de la génération des connecteurs concessifs en dialogue homme-machine

Vladimir Popescu^{1, 2} Jean Caelen¹

(1) GETALP/LIG, Institut National Polytechnique de Grenoble, France

(2) Université « Politehnica » de Bucarest, Roumanie

{vladimir.popescu, jean.caelen}@imag.fr

Résumé. Les connecteurs discursifs ont un rôle important dans l'interprétation des discours (dialogiques ou pas), donc lorsqu'il s'agit de produire des énoncés, le choix des mots qui relient les énoncés (par exemple, en dialogue oral) s'avère essentiel pour assurer la compréhension des *visées* illocutoires des locuteurs. En linguistique computationnelle, le problème a été abordé surtout au niveau de l'interprétation des discours monologiques, tandis que pour le dialogue, les recherches se sont limitées en général à établir une correspondance quasiment biunivoque entre relations rhétoriques et connecteurs. Dans ce papier nous proposons un mécanisme pour guider la génération des connecteurs concessifs en dialogue, à la fois du point de vue discursif et sémantique ; chaque connecteur considéré sera contraint par un ensemble de conditions qui prennent en compte la cohérence du discours et la pertinence sémantique de chaque mot concerné. Les contraintes discursives, exprimées dans un formalisme dérivé de la SDRT (« Segmented Discourse Representation Theory ») seront plongées dans des contraintes sémantiques sur les connecteurs, proposées par l'école genevoise (Moeschler), pour enfin évaluer la cohérence du discours résultant de l'emploi de ces connecteurs.

Abstract. Cue words play an important part in discourse interpretation (whether dialogues are concerned or not), hence when utterances have to be produced, the choice of the words that connect these utterances (for example, in spoken dialogue) is essential for ensuring the comprehension of the illocutionary *goals* of the speakers. In computational linguistics, the issue has been mitigated particularly in the context of monologue discourse interpretation, whereas for dialogues, research is usually limited to establishing an almost one-to-one mapping between rhetorical relations and cue words. In this paper we propose a mechanism for guiding concessive connectors in dialogue, at the same time from a discourse and from a semantic point of view ; each considered connector will be constrained via a set of conditions that take into account discourse coherence and the semantic relevance of each word concerned. Discourse constraints, expressed in a formalism derived from SDRT (“Segmented Discourse Representation Theory”) will be mapped to semantic constraints on the connectors ; these semantic constraints are proposed by the Geneva linguistics school (Moeschler). Finally, the coherence of the discourse resulted from the use of these connectors will be assessed.

Mots-clés : Dialogue homme-machine, cohérence discursive, connecteurs concessifs, sémantique, pragmatique.

Keywords: Human-computer dialogue, discourse coherence, concessive connectors, semantics, pragmatics.

1 Introduction

Les connecteurs discursifs sont importants pour guider la compréhension des messages¹ que les locuteurs transmettent sous forme linguistique en dialoguant. Par conséquent, ils ont reçu relativement beaucoup d'attention dans les recherches en linguistique, surtout lorsque l'interprétation des énoncés est concernée (Ducrot, 1980), (Moeschler, 1989). En traitement automatique des langues, les connecteurs sont regardés en général de deux manières différentes, selon les buts assumés :

- en analyse morpho-syntaxique et sémantique, les connecteurs ont souvent un rôle modéré, car, d'un côté ils subissent et imposent peu de contraintes aux niveaux morphologique et syntaxique (voir par exemple, en français, l'emploi du subjonctif du verbe, après *bien que*) et, d'un autre côté, ils n'apportent rien à la sémantique des énoncés, considérée de manière vériconditionnelle et / ou isolée (Jurafsky & Martin, 2000)² ;
- en interprétation / analyse du discours (qui remonte surtout à la prise en compte des aspects pragmatiques de la langue), les connecteurs ont une place importante, car leur présence guide le processus d'analyse (Mann & Thomson, 1986).

En génération du langage naturel, les connecteurs sont en général considérés comme les manifestations superficielles (c'est-à-dire, au niveau de la réalisation de surface) des relations rhétoriques qui relient des énoncés distincts pour constituer des discours (Reiter & Dale, 2000) ; par exemple, le connecteur "mais" matérialise une relation de *correction* par rapport à ce qui avait été dit auparavant. De toutes façons, cette approche s'avère souvent insuffisante, car, comme l'a montré Moeschler (Moeschler, 1989), les **conditions d'emploi** de chaque connecteur impliquent souvent plusieurs énoncés, reliés de manières assez diverses.

De l'autre côté, au niveau computationnel, les propos de Moeschler doivent être développés sous forme d'algorithmes, où les entrées et les sorties soient clairement précisées. A partir de ce constat, cet article propose une formalisation computationnelle du choix des connecteurs concessifs dans le cadre d'un module générique, conçu pour le contrôle pragmatique de la génération du langage en dialogue oral homme-machine (Popescu *et al.*, 2007).

Au niveau théorique, la nouveauté de l'approche proposée dans cet article réside dans le fait qu'à la différence de Moeschler, qui, étant donné le connecteur, déduisait le schéma sémantico-pragmatique le plus pertinent et donc l'emploi du connecteur (causal ou argumentatif) (Moeschler, 1989), nous proposons de déduire le schéma sémantico-pragmatique le plus pertinent à partir de la structure rhétorique (SDRS - « Segmented Discourse Representation Structure ») du dialogue et, par conséquent, choisir le connecteur qui satisfasse le mieux ce schéma ; ainsi, la cohérence du discours est également prise en compte.

Du point de vue applicatif, une telle démarche est utile lorsqu'il s'agit, pour la machine, de produire des répliques, à partir d'expressions figées à trous, ce qui est une approche courante en dialogue homme-machine finalisé par la tâche, étant données la facilité de l'implémentation et la vitesse d'exécution ; les expressions figées sont ensuite collées via les connecteurs calculés. On génère donc un phénomène de surface (les connecteurs discursifs concessifs) de manière motivée du point de vue pragmatique et sémantique.

¹Le mot « message » est pris ici dans un sens élargi, intégrant la portée spécifiée par la théorie de l'information et les visées illocutoires des sujets parlants.

²Il y a toutefois des exceptions à cette manière de considérer les connecteurs en analyse et en génération, comme par exemple la théorie « sens-texte » de Mel'čuk (Reiter & Dale, 2000).

Pour un état de l'art concis sur ce sujet, on peut retenir, d'un côté, les démarches convergentes avec celles de Moeschler, qui concernent seulement l'*interprétation* des connecteurs du discours ou, d'un autre côté, l'interprétation des connecteurs surtout en termes de relations de discours, soit dans la direction fixée par Marcu (Marcu, 2000), soit selon l'approche de Lascarides en SDRT (Sporelander & Lascarides, 2006). En ce qui concerne le choix des connecteurs en génération, des travaux représentatifs ont été poursuivis par Marcu, qui reliait les connecteurs à des relations discursives seulement (Marcu, 2000) ; donc, selon la terminologie de Moeschler, seul le « cotexte » était pris en compte, en opposition avec le « contexte » sémantico-pragmatique, qui légitime ou bloque la production d'un connecteur, surtout lorsque le dialogue oral est concerné.

Ici, à partir de la structure rhétorique (cotexte), nous calculons un contexte sémantico-pragmatique, qu'on filtre ensuite à travers le cotexte du dialogue entier, et qui pilote le choix des connecteurs. La section suivante constitue le cœur de l'article et présente les sémantiques des connecteurs pragmatiques considérés, aussi que les détails formels et computationnels du mécanisme pilotant le choix des connecteurs en génération ; ce mécanisme sera ensuite évalué de manière qualitative via un exemple typique de dialogue. La troisième section conclut l'article et présente des pistes pour continuer cette recherche.

2 Génération des connecteurs concessifs en dialogue

2.1 Aperçus sémantico-pragmatiques des connecteurs

Les connecteurs discursifs marquant la **concession** ont été bien étudiés, surtout dans l'école genevoise de linguistique (Moeschler, Roulet, Rubattel, etc.) (Moeschler, 1989) ; ces travaux ont abouti à des modélisations sémantiques formelles des connecteurs tels que *mais*, *quand même*, *pourtant* ou *bien que*. Dans ce paragraphe on donne un bref aperçu d'une telle modélisation, à partir de la présentation de (Moeschler, 1989).

Pour cela, on considère d'abord un ensemble de notations, utilisées tout au long de cet article :

- α, β, \dots ::= étiquettes des énoncés dans le discours ;
- $*$::= marque de connexion entre deux énoncés du discours ;
- TRUE, FALSE ::= les valeurs de vérité « vrai », respectivement « faux » ;
- $\text{equals}(\alpha, \varphi)$::= prédicat binaire vrai ssi la variable α est égale (c'est-à-dire, unifiée) avec la valeur φ ;
- $\Rightarrow, \neg, \wedge, \vee$::= connecteurs logiques ayant leur signification classique en logique (Gallier, 1986) ;
- $\in, \exists, \subset, \supset$::= connecteurs ayant leur signification classique en théorie des ensembles ;
- \exists, \forall ::= quantificateurs logiques existentiel, respectivement universel ;
- $K(\alpha)$::= fonction retournant la sémantique (en logique du premier ordre) de l'énoncé α ;
- $v(\lambda)$::= fonction retournant la valeur de vérité de la formule logique λ ; donc, désignant par Λ l'ensemble des formules logiques définissables pour un alphabet donné, on a que $v : \Lambda \rightarrow \{\text{TRUE}, \text{FALSE}\}$;
- $\text{cause}(\alpha, \beta)$::= prédicat binaire vrai ssi la modification de contexte du discours, déterminée par l'énoncé α , est une *cause* suffisante pour l'ajout du contenu de l'énoncé β au contexte ;
- $\rightarrow(\alpha, \beta)$::= relation argumentative entre les énoncés α et β , vraie ssi un locuteur utilise l'énoncé α pour faire admettre le contenu propositionnel où les visées illocutoires de l'énoncé β (Ducrot, 1980) ;
- $<$::= relation de *force argumentative* entre deux énoncés : $\alpha < \beta$ ssi β est plus *pertinent* pour son émetteur (qui tente d'argumenter le contenu d'un autre énoncé, soit-il γ) que α (Moeschler, 1989) ;
- $\nu(\alpha, \beta)$::= relation de *norme*, vraie ssi d'*ordinaire* α implique β ; la norme est donc une implication révisable selon le contexte ;
- $\text{smaller}(\phi, \psi)$::= prédicat binaire vrai ssi la valeur ϕ est inférieure à la valeur ψ ;
- $\text{ClassOf}(\phi, \Phi)$::= prédicat binaire vrai ssi l'élément ϕ appartient à l'ensemble Φ ;

- $\bar{\vee} ::=$ opération logique de « ou exclusif ».

Ainsi, (Moeschler, 1989) définit deux types de concession, **causale** et **argumentative** qui se particularisent pour chacun des connecteurs considérés - *mais*, *quand même*, *pourtant* ou *bien que*. Par conséquent, (Moeschler, 1989) propose un ensemble de sémantiques pour ces quatre connecteurs, dont on ne retient, de toutes façons, que les aspects 'procéduraux', car le but assumé dans cet article est de proposer un cadre *opérationnel* pour la génération de ces connecteurs en dialogue homme-machine.

Ainsi, *mais* et *quand même* se définissent à travers un « carré argumentatif » (délimité par trois énoncés, α , β et γ), formalisable de manière concise comme ci-dessous :

1. *mais* peut être employé de deux manières, désignées par *mais₁* (pour l'emploi de **relation directe**) et *mais₂* (pour l'emploi de **relation indirecte**) :
 - $\alpha \text{ mais}_1 \neg \gamma ::= \exists \beta : \Rightarrow (\alpha, \gamma) \wedge (\gamma \bar{\vee} \neg \gamma) \wedge (\alpha < \beta) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) \wedge \Rightarrow (\beta, \neg \gamma)$;
exemple : Il fait beau *mais₁* je ne sors pas . – α est 'il fait beau', γ est 'je sors' et β est par exemple 'je ne me sens pas bien'.
 - $\alpha \text{ mais}_2 \beta ::= \exists \gamma : \Rightarrow (\alpha, \gamma) \wedge (\gamma \bar{\vee} \neg \gamma) \wedge \Rightarrow (\beta, \neg \gamma) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE})$;
exemple : Il fait beau *mais₂* j'ai mal à la tête. – α est 'il fait beau', β est 'j'ai mal à la tête' et γ est par exemple 'je sors'.
2. *quand même* peut être employé en relation directe seulement, ainsi :
 - $\alpha \text{ quand même } \neg \gamma ::= \exists \beta : \nu(\alpha, \gamma) \wedge (\gamma \bar{\vee} \neg \gamma) \wedge \Rightarrow (\beta, \neg \gamma) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE})$;
 - exemple : Il fait beau. Je reste *quand même* chez moi. – α est 'il fait beau', β est par exemple 'j'ai mal à la tête' et γ est 'je sors' .

D'un autre côté, les connecteurs *pourtant* et *bien que* se définissent à travers un « triangle causal » (délimité par deux énoncés, α et β), formalisable ainsi :

1. *pourtant* peut être employé de deux manières, selon les deux permutations possibles du triangle causal (Moeschler, 1989) :
 - $\alpha \text{ pourtant}_1 \beta ::= \text{cause}(\alpha, \neg \beta) \wedge (\beta \bar{\vee} \neg \beta) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE})$;
exemple : Il pleut, *pourtant₁* je me balade dans le parc. – α est 'il pleut' et β , 'je me balade...'.
 - $\alpha \text{ pourtant}_2 \beta ::= \text{cause}(\beta, \neg \alpha) \wedge (\alpha \bar{\vee} \neg \alpha) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE})$;
exemple : Je suis gentil avec lui, *pourtant₂* il me dit de gros mots toujours. – α est 'je suis gentil...' et β , 'il me dit de gros mots...'.
2. *bien que* peut également être employé de deux manières, selon les permutations du triangle causal :
 - $\alpha \text{ bien que}_1 \beta ::= \text{cause}(\alpha, \neg \beta) \wedge (\beta \bar{\vee} \neg \beta) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE})$;
exemple : *Bien qu'₁* il fasse mauvais, je sors. – α est 'il fait mauvais' et β , 'je sors'.
 - $\alpha \text{ bien que}_2 \beta ::= \text{cause}(\beta, \neg \alpha) \wedge (\alpha \bar{\vee} \neg \alpha) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE})$;
exemple : Je me balade dans le parc, *bien qu'₂* il pleuve. – α est 'je me balade...' et β , 'il pleut'.

Dans ce paragraphe nous avons présenté les sémantiques de quatre connecteurs concessifs, *mais*, *quand même*, *pourtant* et *bien que*, formalisées dans une approche qui s'inspire des travaux de l'école genevoise (Moeschler, 1989). Pour une spécification algorithmique du processus de génération de ces connecteurs, à partir de leurs sémantiques, il y a au moins un point à améliorer dans les sémantiques présentées : on observe qu'au niveau formel il n'y a aucune différence entre les sémantiques de *pourtant* et *bien que*. Comment savoir alors quel connecteur produire, dans un contexte donné ? Pour l'interprétation, Moeschler précise seulement que *bien que* stipule une plus forte détermination causale entre ses arguments, par rapport à *pourtant* (Moeschler, 1989) ; donc, il faudra exprimer cette idée de manière formelle, en tant que règle ou contrainte supplémentaire agissant sur la sémantique de ce connecteur.

2.2 Structure de discours et sémantique des connecteurs

L'idée fondamentale de cet article est de « calculer » le connecteur le plus approprié en génération dialogique, par rapport au contexte sémantico-pragmatique de la conversation. Ainsi, pour cela on part d'une représentation profonde du cotexte associé au dialogue, à laquelle on ajoutera des contraintes contextuelles.

Nous supposons que le point de départ pour le calcul des connecteurs est une structure segmentée de discours (SDRS), calculé en préalable par un module de structuration rhétorique (Popescu *et al.*, 2007). Cette structure de discours est calculée dans le cadre d'une version adaptée de la SDRT, pour la génération en dialogue (Popescu *et al.*, 2007).

Pour qu'on puisse utiliser cette structure rhétorique dans le calcul des connecteurs, nous spécifions une grammaire pour la représenter, utilisant les informations fournies par le module de structuration rhétorique. Une telle grammaire suppose que les informations sur la SDRS consistent en (i) le nombre D des énoncés connectés par des relations rhétoriques, (ii) le nombre R des relations rhétoriques qui relient les énoncés, (iii) l'ensemble Π d'étiquettes π des énoncés, (iv) l'ensemble P des étiquettes des relations rhétoriques³, (v) les formes logiques K des énoncés dans la structure de discours, (vi) les sémantiques (exprimées dans une logique du premier ordre (Popescu *et al.*, 2007)) Σ des relations rhétoriques dans la structure de discours. Cette grammaire est présentée de manière détaillée en (Popescu, 2007).

En ce qui concerne l'ensemble possible des relations rhétoriques qui relient des paires d'énoncés (ou de sous-structures de discours), nous avons considéré 17 relations (Popescu *et al.*, 2007), du total de 35, précisées par la SDRT d'origine (Asher & Lascarides, 2003) : (i) relations **dialogiques** - *Q-Elab* (« Question Elaboration »), *IQAP* (« Indirect Question-Answer Pair »), *P-Corr* (« Plan Correction »), *P-Elab* (« Plan Elaboration »), *Background_q* (« Background Question »), *Elab_q* (« Elaboration Question »), *Narration_q* (« Narration Question »), *QAP* (« Question-Answer Pair »), *ACK* (« Acknowledgement ») et *NEI* (« Not Enough Information ») ; (ii) relations **monologiques** - *Alternation*, *Background*, *Consequence*, *Elaboration*, *Narration*, *Contrast* et *Parallel*. Les sémantiques de toutes ces relations rhétoriques sont simulées dans une logique du premier ordre, pour plus d'efficacité computationnelle dans le contexte des dialogues finalisés par la tâche (Popescu *et al.*, 2007) ; cette simulation conserve les sémantiques informelles des relations, en tant que précisées dans la SDRT d'origine (Asher & Lascarides, 2003).

Pour relier cette structure rhétorique aux spécifications sémantiques des connecteurs concessifs (présentées dans la section 2.1), on va transposer le carré argumentatif et le triangle causal qui situent les quatre connecteurs considérés, dans des contraintes de type rhétorique ; pour cela, on s'appuie sur les sémantiques des relations rhétoriques, en tant que précisées en (Popescu *et al.*, 2007). Ainsi, nous proposons tout d'abord un ensemble de correspondances entre les relations sémantiques spécifiant les connecteurs et les relations rhétoriques ; ces correspondances, présentées dans le tableau 1 sont motivées également par les définitions de (Moeschler, 1989) et par les spécifications de (Asher & Lascarides, 2003).

S'appuyant sur ces correspondances, nous pourrions spécifier les sémantiques des connecteurs, en termes des relations de discours (et des connecteurs logiques usuels). Cependant, une observation préalable s'impose : dans la section 2.1 il n'y avait pas de distinction sémantique formelle entre *pourtant* et *bien que* ; de toutes façons, l'idée d'une plus forte détermination causale agen-

³Les étiquettes des relations rhétoriques sont identiques à leurs noms (Asher & Lascarides, 2003).

TAB. 1 – Relations sémantiques (Moeschler) et relations rhétoriques (Asher & Lascarides)

Relation sémantique	Relation rhétorique
$\bar{\vee}(\alpha, \beta)$	$Contrast(\alpha, \beta) \wedge (\alpha \vee \beta)$
$cause(\alpha, \beta)$	$\alpha \Rightarrow \beta$
$\rightarrow(\alpha, \beta)$	$Consequence(\alpha, \beta)$
$\nu(\alpha, \beta)$	$Background(\alpha, \beta)$

cée par *bien que* peut être formalisée via une contrainte supplémentaire sur ce connecteur : *que le temps de l'action décrite par l'énoncé qui succède ce connecteur précède le temps de l'action décrite par le deuxième énoncé sous l'incidence de ce connecteur*. Au niveau logique cela sera exprimé via la fonction $\Delta(\alpha)$ qui précise (retourne) le temps de l'action décrite par l'énoncé libellé α ; cette fonction est codée dans une **ontologie de discours** (Popescu *et al.*, 2007).

Ainsi, les sémantiques des connecteurs concessifs considérés se formalisent, en termes rhétoriques, comme ci-dessous :

1. $\alpha \text{ mais}_1 \neg\gamma ::= \exists\beta : Consequence(\alpha, \gamma) \wedge Contrast(\gamma, \neg\gamma) \wedge Consequence(\beta, \neg\gamma) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE) \wedge equals(v(K(\gamma)), FALSE) \wedge <(\alpha, \beta)$;
2. $\alpha \text{ mais}_2 \beta ::= \exists\gamma : Consequence(\alpha, \gamma) \wedge Contrast(\gamma, \neg\gamma) \wedge Consequence(\beta, \neg\gamma) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE) \wedge equals(v(K(\gamma)), FALSE)$;
3. $\alpha \text{ quand même } \neg\gamma ::= \exists\beta : Background(\alpha, \gamma) \wedge Contrast(\gamma, \neg\gamma) \wedge Consequence(\beta, \neg\gamma) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE) \wedge equals(v(K(\gamma)), FALSE)$;
4. $\alpha \text{ pourtant}_1 \beta ::= (\alpha \Rightarrow \neg\beta) \wedge Contrast(\beta, \neg\beta) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE)$;
5. $\alpha \text{ pourtant}_2 \beta ::= (\beta \Rightarrow \neg\alpha) \wedge Contrast(\alpha, \neg\alpha) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE)$;
6. $\text{bien que}_1 \alpha, \beta ::= (\alpha \Rightarrow \neg\beta) \wedge Contrast(\beta, \neg\beta) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE) \wedge smaller(\Delta t(\alpha), \Delta t(\beta))$;
7. $\alpha \text{ bien que}_2 \beta ::= (\beta \Rightarrow \neg\alpha) \wedge Contrast(\alpha, \neg\alpha) \wedge equals(v(K(\alpha)), TRUE) \wedge equals(v(K(\beta)), TRUE) \wedge smaller(\Delta t(\beta), \Delta t(\alpha))$.

Dans ces sémantiques, les énoncés non reliés par les connecteurs, mais intervenant dans les contraintes sur ces derniers (γ ou β , selon le cas) sont tirés soit du cotexte, soit du contexte :

- si l'énoncé en question est présent dans le discours, dans des tours de parole antérieurs, on s'appuie sur lui ;
- sinon, on déduit cet énoncé à partir des énoncés antérieurs et de l'**ontologie de tâche** (Caelen & Xuereb, 2007) (le contexte), via un processus inférentiel.

Une dernière remarque à l'égard des sémantiques des connecteurs concessifs (et du lien entre celles-ci et la structure rhétorique) concerne la relation « $< / 2$ », de force argumentative, non formalisée par Moeschler ou dans les sections antérieures de cet article. Pour cela, on s'appuie de manière explicite sur le contexte du *dialogue*, posant qu'on peut avoir deux interlocuteurs, l'émetteur d'un énoncé courant, désigné comme ci-dessus, par la fonction $emitter/1$ et l'allocutaire, désigné par $\neg emitter/1$; par conséquent, il n'y a pas d'équivalence entre des expressions du type $*(\neg emitter(\alpha), \phi)$ et $\neg *(emitter(\alpha), \phi)$, où « $*$ » désigne un prédicat binaire quelconque et « ϕ », un terme quelconque. Ceci étant, nous posons comme hypothèse de travail qu'un énoncé dont le contenu fait une référence explicite à son émetteur a une force argumentative plus élevée par rapport à un énoncé qui ne le fait pas, ou qu'un énoncé qui fait une référence explicite à l'allocutaire (donc, au destinataire) a une force argumentative plus élevée par rapport à un énoncé dont le contenu ne fait référence ni à cet allocutaire, ni à l'émetteur. Précisant que deux énoncés dont la force argumentative est à comparer ont le même émetteur, ces consignes peuvent s'exprimer sous forme logique ainsi :

$$\alpha < \beta ::= (ClassOf(emitter(\beta), K(\beta)) \wedge \neg ClassOf(emitter(\alpha), K(\alpha))) \vee (ClassOf(\neg emitter(\beta), K(\beta)) \wedge \neg ClassOf(\neg emitter(\alpha), K(\alpha))) \wedge \neg ClassOf(emitter(\alpha), K(\alpha)) \wedge equals(emitter(\alpha), emitter(\beta)) \wedge equals(\neg emitter(\alpha), \neg emitter(\beta)).$$

2.3 Choix des connecteurs : algorithme

Le cadre posé dans le paragraphe précédent pour l'expression sémantique, amendée par la structure rhétorique, des connecteurs concessifs en dialogue sera mis en valeur via une procédure conçue pour générer les connecteurs appropriés, lorsqu'il s'agit, pour la machine, de produire un énoncé dialogique.

Les entrées dans cette procédure consistent en (i) la structure rhétorique du dialogue « courant » (c'est-à-dire, intégrant aussi la sémantique de la réplique censée être générée par la machine et donc, éventuellement, contenir un ou plusieurs connecteurs concessifs), (ii) les sémantiques des connecteurs (spécifiées sous forme de règles), (iii) l'ontologie de discours (spécifiée sous forme d'arbre hiérarchique (Popescu *et al.*, 2007)) et (iv) l'ontologie de tâche pour les dialogues (spécifiée, à l'instar de l'ontologie de discours, sous forme de hiérarchie de concepts (Caelen & Xuereb, 2007)).

La sortie de cette procédure consiste en un ensemble de connecteurs qui peuvent relier les énoncés dans la réplique courante de la machine ; cette liste peut éventuellement être vide, mais dans le cas contraire, chaque entrée dans cette liste a trois éléments : (i) le connecteur choisi, (ii) son « argument gauche » et (iii), son argument droit ; donc une entrée type est de la forme : $(*, \pi_i, \pi_j)$, ce qui correspondra à une forme de surface du type $\pi_i * \pi_j$, où $*$ est le connecteur et π_i et π_j , les libellés de deux énoncés. Donc, la procédure est spécifiée ci-dessous :

1. **initialiser** la liste des connecteurs : $\Lambda \leftarrow \emptyset$;
2. **lire** la structure de discours courante, $S(D, R, P, \Pi)$ (cf. §2.2) ;
3. **lire** les étiquettes des énoncés à être générés par la machine, $\langle \pi \rangle$, avec $\langle \pi \rangle \subset \Pi$ et $|\langle \pi \rangle| < D$;
4. **trier** l'ensemble $\langle \pi \rangle$ par rapport à l'ordre d'apparition des énoncés (ordre spécifié par le composant de structuration rhétorique (Popescu *et al.*, 2007)) ;
5. **pour** chaque énoncé π_- dans $\langle \pi \rangle$:
 - (a) **repérer** les relations rhétoriques connectant l'énoncé π_- à des énoncés dans S ; soient ces relations spécifiées par l'ensemble $\langle \rho \rangle_{\pi_-} \subset P$, avec $|\langle \rho \rangle_{\pi_-}| < R$;
 - (b) **repérer** les arguments *gauches* des relations dans l'ensemble $\langle \rho \rangle_{\pi_-}$; soient ces arguments spécifiés par l'ensemble $\langle \gamma \rangle_{\pi_-} \subset \Pi$, avec $|\langle \gamma \rangle_{\pi_-}| < D$;
 - (c) **pour** chaque sémantique ξ des connecteurs ($\xi \in \Xi$, avec $|\Xi| = 7$, car il y a 7 connecteurs possibles) :
 - i. **appairier**($\xi, \pi_-, (\langle \rho \rangle_{\pi_-}, \langle \gamma \rangle_{\pi_-})$) ;
 - ii. **si** OK, **alors** $\Lambda \leftarrow \Lambda \cup (\xi, \pi^\xi, \pi_-)$;
 - iii. **sinon**, **alors** $\Lambda \leftarrow \Lambda \cup \emptyset$;
6. **si** $|\langle \pi^\xi, \pi_- \rangle| > 1$ (la procédure **appairier** retourne plusieurs connecteurs entre deux énoncés), **alors** :
 - (a) **s'il** existe un connecteur qui a un maximum de contraintes, **alors garder** celui-ci et **enlever** le reste (par exemple, entre *pourtant* et *bien que*, garder le dernier et enlever le premier) ;
 - (b) **sinon**, **s'il** existe un connecteur qui assure un nombre maximal de relations rhétoriques, c'est-à-dire, qui contient un nombre maximal de relations rhétoriques *différentes* dans ses contraintes, **alors** :
 - i. **si** les sémantiques des connecteurs ne mènent pas à une contradiction logique, **alors garder** tous ces connecteurs ;
 - ii. **sinon**, **alors garder** celui-ci et **enlever** le reste (par exemple, entre *mais* et *quand même*, garder le dernier et enlever le premier, car celui-ci contient deux relations rhétoriques différentes, *Consequence* et *Contrast*, tandis que le dernier en contient trois - les deux qui viennent d'être précisées, auxquelles *Background* s'ajoute) ;
 - (c) **sinon**, **alors choisir** de manière aléatoire un des connecteurs (cette situation est peu probable, étant données les sémantiques des connecteurs et leur utilisation) ;
7. **retourner** Λ .

Le noyau de l'algorithme consiste dans la procédure **appairier**/3, qui prend en entrée la sémantique d'un connecteur, (l'étiquette d') un énoncé qui constituera l'argument *droit* du connecteur et une structure formée de deux ensembles : un ensemble de relations rhétoriques (censées avoir π_- en tant qu'argument droit) et, respectivement, un ensemble d'énoncés (censés être les arguments gauches des relations rhétoriques). La sortie de la procédure **appairier** est constituée d'une structure qui contient la sémantique d'un connecteur, son argument gauche et, respectivement, son argument droit (qui est censé être justement l'énoncé π_- lu en entrée) :

1. **initialiser** trois listes : $\bar{\Pi} \leftarrow \emptyset$, $\bar{P} \leftarrow \emptyset$, $\bar{C} \leftarrow \emptyset$; ces listes contiendront, respectivement, des étiquettes d'énoncés, des étiquettes de relations rhétoriques, et des connecteurs logiques ;
2. **pour** chaque terme dans la sémantique ξ du connecteur, le **classer** dans une des catégories suivantes : (i) étiquettes d'énoncés, (ii) relations de discours, (iii) connecteurs logiques ;
3. par conséquent, **actualiser** les listes $\bar{\Pi}$, \bar{P} et \bar{C} ;
4. **si** $\bar{P} \subset < \rho >_{\pi_-}$, **alors** :
 - (a) **unifier** les termes dans $\bar{\Pi}$ qui sont les arguments des termes en \bar{P} , avec les termes en $< \gamma >_{\pi_-}$ qui sont les arguments des termes en $\bar{P} \cap < \rho >_{\pi_-}$;
 - (b) **pour** chaque connecteur logique en \bar{C} , **vérifier** si celui-ci peut être placé entre des entités en $< \rho >_{\pi_-} \cap \bar{P}$ et $< \gamma >_{\pi_-} \cap \bar{\Pi}$, afin que :
 - la succession des termes et connecteurs soit identique à celle spécifiée en ξ , à part les unifications de l'étape 4.(a) ;
 - la forme logique résultée (en étendant les énoncés à leurs sémantiques, exprimées via des prédicats dans l'ontologie de tâche, selon les détails précisés en (Popescu *et al.*, 2007)) ne mène pas à l'ensemble vide (c'est-à-dire, à une contradiction) ;
 cette étape représente en fait la résolution de la formule logique obtenue de ξ , en substituant les énoncés par des énoncés dans la structure de discours en entrée, pourvu qu'on préserve les arguments des relations rhétoriques ;
 - (c) **si** OK, **alors retourner** OK, (ξ, π^ξ, π_-) ;
 - (d) **sinon, alors retourner** KO ;
5. **sinon, alors retourner** KO.

2.4 Traçage sur une situation de dialogue

Nous démontrons l'opération des algorithmes proposés en §2.3 sur un fragment de dialogue typique entre le système et un client, où la machine est une bibliothécaire virtuelle. Ainsi, désignant par U le client (l'utilisateur), par M , la machine, et par l'indice supérieur π_{ij} , l'étiquette du j -ème énoncé dans le i -ème tour de parole, le dialogue est montré ci-dessous (les connecteurs ont été mis en italiques) :

U : Excusez-moi ^{π_{11}} , pourrais-je emprunter ce bouquin ? ^{π_{12}}

M : Monsieur, vous ne pouvez emprunter que trois livres à la fois ^{π_{21}} , *mais* je ferai *quand même* une exception cette fois-ci. ^{π_{22}}

U : Merci beaucoup ! ^{π_{31}} Alors, je vous donne ma carte ? ^{π_{32}}

M : Bon ^{π_{41}} , *bien que* vous ayez déjà gardé un livre il y a deux mois ^{π_{42}} , je vous en prête maintenant quatre ^{π_{43}} ; *pourtant*, vous n'avez pas l'air plus attentif en ce moment... ^{π_{44}}

Les énoncés de M sont disponibles sous forme logique, en tant que fournis par le contrôleur de dialogue (Caelen & Xuereb, 2007), (Popescu *et al.*, 2007), tandis que ceux de U , sous forme linguistique ; à partir de ces derniers, l'analyseur sémantique calcule, pour ces énoncés, les formes logiques correspondantes, auxquelles l'interpréteur pragmatique attache les informations concernant le type d'acte de langage véhiculé par chaque contribution dialogique de

l'utilisateur (Caelen & Xuereb, 2007). A partir de ces informations, le composant de structuration rhétorique dans le module de génération calcule la SDRS qui intègre les formes logiques des énoncés censés être réalisés sous forme linguistique par M (Popescu *et al.*, 2007). Ainsi, pour le fragment de dialogue considéré, la SDRS calculée est exprimée, sous forme synthétique, ci-dessous :

$$\text{Elaboration}(\pi_{11}, \pi_{12}) \wedge P\text{-Corr}(\pi_{12}, \pi_{21}) \wedge \text{Consequence}(\pi_{21}, \neg\pi_{22}) \wedge \text{ACK}(\pi_{22}, \pi_{31}) \wedge \text{Consequence}(\pi_{31}, \pi_{32}) \wedge \text{ACK}(\pi_{32}, \pi_{41}) \wedge P\text{-Elab}(\pi_{22}, \pi_{43}) \wedge \text{Consequence}(\pi_{42}, \neg\pi_{43}) \wedge \text{Consequence}(\pi_{43}, \neg\pi_{44}) \wedge \text{Narration}(\pi_{42}, \pi_{44}).$$

Les relations de discours ont été calculées entre des paires d'énoncés et même entre des paires dont un élément est un énoncé sous forme niée ; cela est utile pour faire la correspondance entre des fragments de cette SDRS et les sémantiques des connecteurs. Ainsi, pour la première réplique de la machine, entre les énoncés π_{21} et π_{22} la procédure **appariar** retourne les éléments (*mais*, π_{21} , π_{22}) et (*quand même*, π_{21} , π_{22}), puisque, pour *mais* on retrouve les contraintes $\text{Consequence}(\pi_{21}, \neg\pi_{22}) \wedge \text{Contrast}(\pi_{22}, \neg\pi_{22}) \wedge \text{equals}(v(K(\pi_{21})), \text{TRUE}) \wedge \text{equals}(v(K(\neg\pi_{21})), \text{FALSE})$ et l'énoncé « manquant » β est retrouvé en tant que 'vous ne pouvez pas emprunter ce livre' qui est déduit par *modus ponens* (Gallier, 1986) à partir de π_{21} et du fait (connu par le système à travers son ontologie de tâche) que U a déjà emprunté trois livres qu'il n'a pas encore rendus ; pour *quand même* est déduit via les mêmes contraintes que ci-dessus, auxquelles s'ajoute la contrainte $\text{Background}(\pi_{21}, \neg\pi_{22})$, déduite par *modus ponens* du fait que U a déjà emprunté trois livres et que, par défaut, selon le règlement, la bibliothèque ne prête pas plus de trois livres à la fois. Dans ce contexte, on a trois choix : (i) supprimer *mais*, (ii) supprimer *quand même*, (iii) garder les deux connecteurs ; selon l'algorithme dans §2.3 nous devrions adopter un des choix (ii) ou (iii). Du fait que (iii) ne mène à aucune contradiction logique, nous l'adoptons, car ainsi nous augmentons la pertinence *illocutoire* de la contribution langagière de M . De toutes façons, pour pouvoir garder *mais* on doit vérifier la satisfaction de $< (\pi_{21}, \beta)$, ce qui est vrai puisque π_{21} exprime une vérité générale - 'on peut emprunter au plus trois livres à la fois', tandis que β exprime un fait relié à U , donc : $\text{ClassOf}(\neg\text{emitter}(\beta), K(\beta))$.

La deuxième contribution de la machine est traitée de manière similaire : entre les énoncés π_{42} et π_{43} la procédure **appariar** retourne (*bien que*, π_{42} , π_{43}) et (*pourtant*, π_{42} , π_{43}), puisqu'on retrouve la contrainte $\pi_{42} \Rightarrow \neg\pi_{43}$, déduite de la relation $\text{Consequence}(\pi_{42}, \neg\pi_{43})$ et du fait, présent dans l'ontologie de tâche, que si un client avait perdu un document, la bibliothèque serait devenue plus stricte avec celui-ci, ce qui n'est pas compatible avec la concession de prêter plus de documents que le règlement prévoit, à cet usager (quatre livres, au lieu de trois) ; toutefois, on en retient seulement *bien que*, car, selon l'algorithme dans §3.2, ce connecteur est défini par plus de contraintes : à partir de $K(\pi_{42})$ et de $K(\pi_{43})$ (les sémantiques des énoncés π_{43} et π_{44}) on déduit $\text{smaller}(\Delta t(\pi_{42}), \Delta t(\pi_{43}))$. Par contre, pour les énoncés π_{43} et π_{44} , la procédure **appariar**, opérant à l'instar du cas précédent, ne retourne quand même que *pourtant*, car, du fait que le temps de ces deux énoncés est le même (le présent), on a que $\text{equals}(\Delta t(\pi_{43}), \Delta t(\pi_{44}))$, donc les contraintes pour *bien que* ne sont pas satisfaites.

3 Conclusion

Dans cet article nous avons proposé un cadre pour contrôler la génération des connecteurs concessifs afin de mieux structurer les répliques de la machine en dialogue avec les utilisateurs. Le mécanisme proposé pour générer *mais*, *quand même*, *pourtant* et *bien que* s'articule d'une part autour d'une description sémantique de ces connecteurs, dérivée de celle proposée

par (Moeschler, 1989) et, d'une autre part, autour d'un ensemble de spécifications pragmatiques dans une procédure computationnelle. Ainsi, l'efficacité de ce cadre a été démontrée sur un fragment typique de dialogue homme-machine autour d'une application particulière - la réservation de documents dans une bibliothèque. Toutefois, les démarches proposées sont génériques, voire indépendantes du domaine d'application et paramétrables par rapport à celui-ci.

Cette recherche devrait être continuée à plusieurs niveaux : (i) l'ensemble des connecteurs devrait être étendu à d'autres, comme *parce que*, *puisque*, *car*, *d'ailleurs*, etc. ; (ii) le mécanisme devrait être étendu au cas du dialogue à plusieurs locuteurs, ce qui reviendrait dans un premier temps à la modification de la sémantique de la relation de force argumentative $<$: les interlocuteurs de la machine ne sont plus réduits à $\neg\text{emitter}()$; du fait que plusieurs interlocuteurs interviennent, on peut avoir un ensemble plus large de situations concernant la force argumentative : par exemple un énoncé dont l'action implique un locuteur plus haut du point de vue hiérarchique ou plus actif en conversation peut être plus pertinent du point de vue argumentatif qu'un autre énoncé impliquant son émetteur... ; (iii) une fois réalisées les extensions au dialogue à plusieurs, ce mécanisme devra être évalué de manière statistique sur des dialogues réels.

Références

- ASHER N. & LASCARIDES A. (2003). *Logics of Conversation*. United Kingdom : Cambridge University Press.
- CAELEN J. & XUEREBA A. (2007). *Interaction et pragmatique - jeux de dialogue et de langage*. Paris : Hermès Science.
- DUCROT O. (1980). *Les mots du discours*. Paris : Editions du Minuit.
- GALLIER J. (1986). *Logic for Computer Science*. New York : Wiley.
- JURAFSKY D. & MARTIN J. H. (2000). *Speech and Language Processing*. New Jersey : Prentice Hall.
- MANN W. C. & THOMSON S. A. (1986). *Rhetorical Structure Theory : description and construction of text structures*. Nijmegen, The Netherlands : Information Sciences Institute.
- MARCU D. (2000). *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. Canada : Ph D Thesis, University of Toronto.
- MOESCHLER J. (1989). *Modélisation du dialogue - représentation de l'inférence argumentative*. Paris : Hermès Science.
- POPESCU V. (2007). Pronominal anaphora generation in human-computer dialogue, using a semantic-pragmatic approach. In *Proceedings of the Doctoral Consortium at the FoLLI Summer School EUROLAN*, Iasi, Romania : "Al. I. Cuza" University Press.
- POPESCU V., CAELEN J. & BURILEANU C. (2007). First-order logic rhetorical structuring for natural language generation in human-computer dialogue. In *Lecture Notes in Computer Science 4629*, Heidelberg : Springer.
- REITER E. & DALE R. (2000). *Building Natural Language Generation Systems*. United Kingdom : Cambridge University Press.
- SPORELANDER C. & LASCARIDES A. (2006). Using automatically labelled examples to classify rhetorical relations : an assessment. In *Journal of Natural Language Engineering*, United Kingdom : Cambridge University Press.