

Réseau bayésien pour un modèle d'utilisateur et un module de compréhension pour l'optimisation des systèmes de dialogues

Olivier Pietquin

Supélec, Campus de Metz – Equipe STS
2 rue Edouard Belin – F-57070 Metz
olivier.pietquin@supelec.fr

Mots-clés : Systèmes de dialogue, simulation de dialogues, modèle d'utilisateur, optimisation.

Keywords: Spoken dialog systems, dialog simulation, user modeling, optimization

Résumé Dans cet article, un environnement modulaire pour la simulation automatique de dialogues homme-machine est proposé. Cet environnement comprend notamment un modèle d'utilisateur consistant dirigé par le but et un module de simulation de compréhension de parole. Un réseau bayésien est à la base de ces deux modèles et selon les paramètres utilisés, il peut générer un comportement d'utilisateur cohérent ou servir de classificateur de concepts. L'environnement a été utilisé dans le contexte de l'optimisation de stratégies de dialogue sur une tâche simple de remplissage de formulaire et les résultats montrent qu'il est alors possible d'identifier certains dialogues problématiques du point de vue de la compréhension.

Abstract In this paper we present a modular environment for simulating human-machine dialogues by computer means. This environment includes a consistent goal-directed user model and a natural language understanding system model. Both models rely on a special Bayesian network used with different parameters in such a way that it can generate a consistent user behaviour according to a goal and the history of the interaction, and been used as a concept classifier. This environment was tested in the framework of optimal strategy learning for the simple form-filling task. The results show that the environment allows pointing out problematic dialogues that may occur because of misunderstanding between the user and the system.

1 Introduction

Dans cet article, nous traitons essentiellement de simulation de dialogues homme-machine. Initialement, les systèmes de simulation étaient destinés essentiellement à la validation de modèles du discours (Power, 1979). Avec l'apparition des interfaces vocales sont aussi arrivés les problèmes de conception. La conception de ces interfaces est un processus cyclique dans lequel interviennent successivement des phases de développement, de tests, d'évaluations et d'améliorations. La phase la plus sujette aux contraintes de temps et d'argent et bien souvent celle de l'évaluation et de test. Pour cette raison, la simulation en vue de l'évaluation automatique des interfaces s'est répandue depuis la fin des années 1990 (Eckert et al., 1998). De cette combinaison de la simulation et de l'automatisation de l'évaluation a assez vite découlé une nouvelle application : l'apprentissage automatique de stratégies optimales (Levin, Pieraccini, 1997) (Singh et al., 1999). Dans cet article, un environnement de simulation de dialogues est proposé dans le cadre de cette dernière application.

De tels environnements existent donc déjà. Certains utilisent des modèles statistiques de transitions entre états obtenus d'après observation de dialogues réels, (Singh et al., 1999). D'autres utilisent un modèle d'utilisateur sans mémoire (Levin, Pieraccini, 1997) et n'incluent pas de modélisation de l'erreur. Ici, nous décrivons un environnement de simulation comprenant un modèle d'utilisateur consistant étant donné l'historique de l'interaction (avec mémoire) et un but. Cet environnement comprend aussi un modèle de système de reconnaissance vocale ainsi qu'un module simulant la compréhension du langage naturel. En incluant ces modules dans l'environnement, nous espérons que les stratégies apprises tiendront comptes de leurs lacunes.

2 Un modèle formel pour le dialogue vocal homme-machine

De manière formelle et comme le décrit la Figure 1, un dialogue vocal homme-machine peut être considéré comme un processus séquentiel dans lequel un utilisateur humain et un système de gestion de dialogue (DM : *Dialogue Manager*) communiquent grâce à la parole au travers d'un canal de transmission. Ce canal est composé de différents modules qui manipulent chacun l'information pour lui faire prendre une forme utilisable par le ou les modules suivants. Le but d'un système de dialogue étant souvent de fournir de l'information à l'utilisateur, le système de gestion de dialogues peut donc accéder à une base de connaissance.

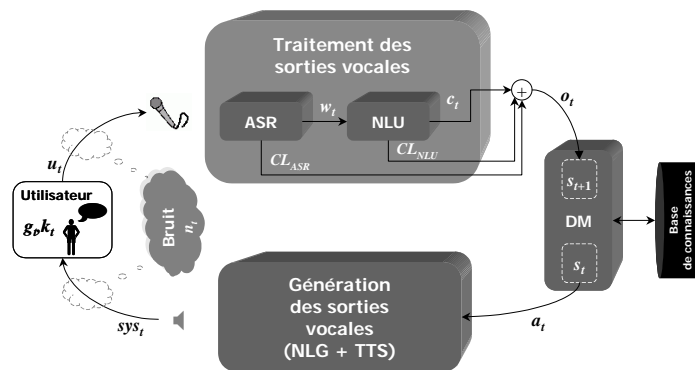


Figure 1 : Modèle de dialogue vocal homme-machine

Le processus étant séquentiel, il peut être discrétisé en *tours* t . À chaque tour, le gestionnaire de dialogue génère un ensemble d'*actes de communication* a_t sur base de son *état* interne s_t pouvant se matérialiser en une invite, une question, une aide, une demande de confirmation, la fermeture du dialogue etc. Afin d'être compris par l'utilisateur, cet ensemble est transformé en un signal de parole sys_t par

les systèmes de génération de sorties vocales. En fonction de ce qu'il a pu comprendre de ce signal, de sa

connaissance au moment t (k_t) et du *but* qu'il poursuit en communiquant avec le système (g_t), l'utilisateur produit à son tour un signal de parole u_t . Dans le cas particulier des systèmes de dialogue, le terme 'connaissance' peut faire référence à la connaissance de l'utilisateur concernant l'*historique* de l'interaction, la *tâche*, le *système* lui-même ou le *monde* en général. Les deux signaux vocaux u_t et sys_t sont entachés par le *bruit* ambiant n_t au moment de leur production. Le système de reconnaissance vocale (ASR) traite alors le signal u_t et le transforme en un ensemble de *mots* w_t . Au passage, le module ASR produit une mesure CL_{ASR} indiquant le degré de *confiance* qu'il accorde à son résultat. L'ensemble w_t est ensuite passé au système de compréhension de parole (NLU) qui doit en retirer une représentation sémantique que nous supposons mise sous la forme d'un ensemble de *concepts* c_t . Le module NLU produit lui-aussi une mesure de confiance CL_{NLU} associée à l'ensemble c_t . L'ensemble $\{c_t, CL_{ASR}, CL_{NLU}\}$ compose une *observation* o_t qui est utilisée pour réaliser une mise à jour de son état interne. D'un point de vue probabiliste, le comportement de l'utilisateur peut être résumé par la probabilité conjointe suivante :

$$\begin{aligned}
 P(u, g, k | sys, a, s, n) &= \underbrace{P(k | sys, a, s, n)}_{\text{MAJ de connaissance}} \cdot \underbrace{P(g | k, sys, a, s, n)}_{\text{Modification du but}} \cdot \underbrace{P(u | g, k, sys, a, s, n)}_{\text{Sortie utilisateur}} \\
 &= \underbrace{P(k | sys, s, n)}_{\text{MAJ de connaissance}} \cdot \underbrace{P(g | k)}_{\text{Modification du but}} \cdot \underbrace{P(u | g, k, sys, n)}_{\text{Sortie utilisateur}}
 \end{aligned} \tag{1}$$

Les simplifications dans (1) tiennent compte de plusieurs faits, notamment on peut raisonnablement admettre que la connaissance de l'utilisateur n'est pas modifiée par l'acte a puisque l'utilisateur n'a pas accès directement à cette valeur. De même, sa réponse ne dépend ni de l'acte a qu'il ne connaît

pas, ni de l'état s qu'il a du intégrer dans sa connaissance de l'historique de l'interaction. Enfin, une modification du but de l'utilisateur doit passer par une modification de sa connaissance uniquement. Les trois termes de (1) mettent en évidence les relations étroites qui existent entre le processus de production de parole et le couple {but, connaissance}. Néanmoins, la modification de la connaissance est un processus incrémental (mise à jour) et se base donc aussi sur la connaissance préalable de l'utilisateur :

$$\begin{aligned} P(k | sys, s, n) &= \sum_{k^-} P(k | k^-, sys, s, n) \cdot P(k^- | sys, s, n) \\ &= \sum_{k^-} P(k | k^-, sys, n) \cdot P(k^- | s) \end{aligned} \quad (2)$$

Ici, k^- représente la variable k_{t-1} . La simplification du second facteur de la somme provient du fait évident que la connaissance de l'utilisateur au temps $t-1$ ne peut pas dépendre des signaux de parole ou de bruit au temps t .

3 Le modèle d'utilisateur

3.1 Un réseau bayésien dynamique

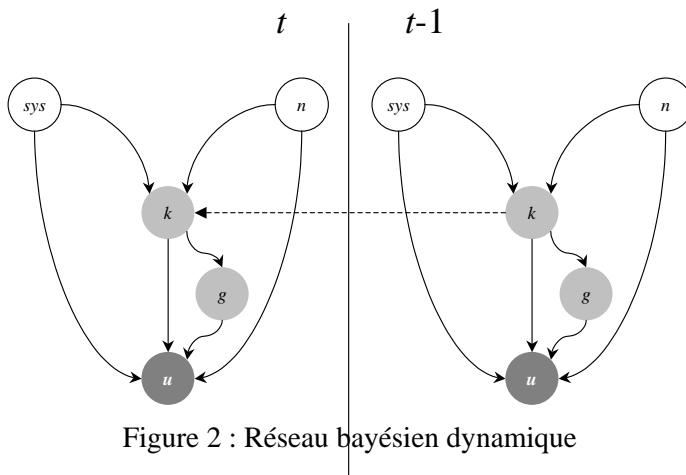


Figure 2 : Réseau bayésien dynamique

(cercles vides), les variables k et g sont des variables internes (cercles gris-clair) et la variable u est une variable de sortie (cercles gris-foncé).

Les équations (1) et (2) permettent de dire qu'un réseau bayésien dynamique (DBN : Dynamic Bayesian Network) pourrait encoder la factorisation particulière des probabilités associées à l'utilisateur (Pearl, 1988). Les nœuds du réseau sont donnés par les variables présentes dans les équations (sys, n, k, g, u) et les arcs sont donnés par les probabilités conditionnelles. La consistance de tour en tour est assurée par la dépendance dans le temps de la variable k . Le réseau dynamique obtenu est montré sur la Figure 2. Les variables sys et n sont des variables extérieures à l'utilisateur

3.2 Utilisation du Modèle

Le DBN de la Figure 2 paraît relativement simple, néanmoins la définition des variables qu'il fait intervenir est plus ou moins floue. Ici, nous avons choisi une représentation des variables en paires « attribut-valeur » (paires AV) dérivées de la description en « Matrice attribut-valeur » de la tâche. Dans ce cadre, chaque acte de communication est considéré comme un ensemble de paires AV. Dans ce qui suit, Le signal de parole sys émis par le système est alors modélisé par un ensemble de paires AV dont l'ensemble des attributs, noté $S = \{s^\sigma\}$, contient des éléments qui peuvent prendre des valeurs booléennes indiquant si oui ou non l'attribut associé est présent dans sys . Un attribut spécial non booléen A_S sera inclus à S et sa valeur définira le type d'acte de communication associé à sys . Les types acceptés peuvent être 'invite', 'question', 'demande de relaxation', 'proposition', 'demande de confirmation', 'fermeture du dialogue', ... Une question directe sera alors caractérisée par un attribut A_S égal à 'question' et un seul attribut s^σ dont la valeur sera vraie. La réponse u de l'utilisateur sera modélisée par une autre paire AV dans laquelle les attributs appartiennent à $U = \{u^\nu\}$ et l'ensemble des valeurs possibles pour chaque attribut u^ν sera noté $V = \{v_i^\nu\}$. Un attribut spécial C_U est ajouté à U et sa valeur booléenne indique si l'utilisateur a décidé de clore le dialogue dans sa réponse. Le but et la

connaissance de l'utilisateur seront représentées respectivement par les paires $\mathbf{G} = \{[g^\gamma, gv_i^\gamma]\}$ et $\mathbf{K} = \{[k^\kappa, kv_i^\kappa]\}$ ou g^γ et k^κ sont des attributs et gv_i^γ et kv_i^κ sont les valeurs possibles. En fonction de ces nouvelles notations, le réseau de la Figure 2 devient celui de la Figure 3 ou la dépendance dans le temps a été volontairement omise pour plus de clarté ainsi que le bruit dont la modélisation est trop complexe. Chaque valeur ou *état* possible pour chaque variable de ce réseau est une combinaison des attributs et des valeurs, ce qui signifie que les états sont discrets et en nombre fini. On peut donc définir une version factorisée de ce réseau dans laquelle figureraient les variables $A_s, s^\sigma, v_i^\sigma, u^\nu, v_i^\nu, g^\gamma, gv_i^\gamma, k^\kappa, kv_i^\kappa$ et U_C .

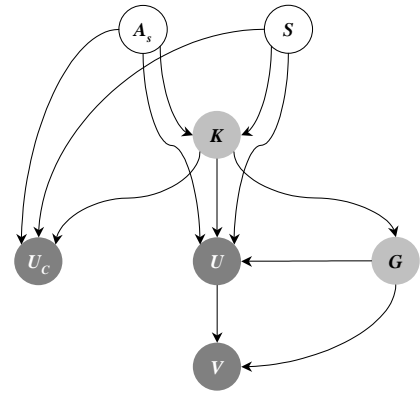


Figure 3 : Réseau bayésien basé sur les paires AV

Considérons une tâche simple consistant à remplir un formulaire composé de deux entrées : $S = \{s^1, s^2\}$. Le système peut utiliser 4 types d'actes de communication : 'invite', 'question directe', 'demande de confirmation' et 'fermeture'. Pour simplifier, considérons que la connaissance de l'utilisateur se compose de simples compteurs, chacun associé à un élément de S , initialisés à 0 et qui sont incrémentés à chaque fois que le système pose une question ou demande une confirmation sur l'entrée associée. Ceci est suffisant pour permettre au modèle d'utilisateur de rester consistant par rapport à l'historique de l'interaction et de réagir à un comportement insatisfaisant du système (en réagissant lorsqu'une entrée a été demandée plusieurs fois). Le but de l'utilisateur est alors de transmettre au système les valeurs correctes pour les attributs représentés par les entrées du formulaire (Figure 4).

Goal		Know.
Att.	Val.	Count
g^1	gv_1	k^1
g^2	gv_2	k^2

Figure 4 : But et connaissance de l'utilisateur

L'utilisateur peut donc inclure dans ses réponses u les deux attributs u^1 et u^2 (il y a autant d'attributs dans U que dans S). Afin de simuler la réponse de l'utilisateur à l'invite, il suffit alors d'entrer l'évidence suivante dans le moteur d'inférence :

A_s	k^1	k^2	g^1	g^2	gv^1	gv^2
invite	0	0	1	1	gv_1	gv_2

Figure 5 : Evidence pour une réponse à l'invite

Les valeurs 1 associées aux variables g^i signifient que les attributs g^i sont bien présents dans le but. Grâce à cette évidence, le moteur d'inférence produira les probabilités $P(u^1=1)$, $P(u^2=1)$, $P(U_C=1)$ et leurs compléments. Tout d'abord, le modèle choisit de manière aléatoire un nombre réel entre 0 et 1, si ce nombre est inférieur à $P(U_C=1)$, le dialogue est clos. Dans le cas contraire, le même processus est répété pour choisir les attributs présents dans la réponse de l'utilisateur. En supposant que u^1 est sélectionnée pour être présente dans la réponse de l'utilisateur, l'évidence suivante est alors entrée dans le moteur d'inférence :

u^1	u^2	gv^1	gv^2
1	0	gv_1	gv_2

Figure 6 : Inférence pour une valeur de réponse

4 Simulation de la compréhension de parole

La simulation de NLU peut se faire en utilisant le réseau bayésien décrit plus haut comme classificateur. Pour ce faire, nous considérerons que les erreurs de reconnaissances vocales n'affectent que les valeurs des paires AV alors que les erreurs d'associations attribut-valeur sont dues au module

de compréhension. En considérant que le processus de reconnaissance vocale a transformé les valeurs $V = \{v_i^v\}$ générées dans sa réponse u par le modèle d'utilisateur en un ensemble de valeur $W = \{w_j\}$ et en reprenant l'exemple simple du remplissage de formulaire expliqué dans la section précédente, les évidences suivantes peuvent être introduites dans le moteur d'inférence pour simuler la compréhension de la réponse à l'invite :

A_S	s^1	s^2	with	v_1^1	or	v_2^1
invite	0	0		w_j		w_j

Figure 7 : Evidence pour la compréhension de la réponse à l'invite

A moins que w_j ne soit pas une valeur acceptable pour un des attributs testés, ces deux différentes évidences vont fournir des valeurs pour les probabilités $P(u^1 | A_S = greet, v_1^1 = w_j)$ and $P(u^2 | A_S = greet, v_2^1 = w_j)$. Le système de simulation de compréhension va alors affecter la valeur w_j à l'attribut u^i ayant produit la probabilité la plus haute. Des situations plus complexes peuvent évidemment être rencontrées mais il est toujours possible de les transformer en évidence utilisable par le moteur d'inférence. Cette méthode peut aussi produire une sorte de niveau de confiance de compréhension. Dans le cas de la classification d'une seule valeur, le niveau de confiance de compréhension est simplement la probabilité fournie par le moteur d'inférence. Lorsque plusieurs valeurs ont du être associée à des attributs par le module de compréhension, une mesure de confiance peut être affectée à chaque paire ou une mesure globale peut être donnée en multipliant toutes les valeurs.

5 Apprentissage de stratégies optimales par simulation

Le modèle décrit ci-dessus a été développé dans le but de l'apprentissage automatique de stratégies de dialogue homme-machine optimales. Nous avons donc mis notre environnement en présence d'un agent d'apprentissage par renforcement comme proposé dans (Levin, Pieraccini, 1997). Pour se faire, il faut définir un critère d'optimisation. On peut en trouver plusieurs dans la littérature néanmoins, l'hypothèse selon laquelle la contribution de chaque acte à la satisfaction de l'utilisateur est une bonne mesure de l'évaluation d'une stratégie est retenue ici. Selon (Singh et al, 1999) une fonction de coût basée sur une mesure de la complétion de la tâche, les performances de reconnaissance et de compréhension et la durée en tours du dialogue serait satisfaisante. Dans notre expérience, les utilisateurs sont invités à fournir des informations à propos d'un voyage en train. Les attributs sont donc une ville de départ, une ville de destination, une heure de départ, une heure d'arrivée désirée et la classe. Il y a 50 valeurs possibles pour les villes (les mêmes pour le départ et l'arrivée) et les heures possibles sont les heures plaines (de 0 à 24). Les types d'actes de communications possibles sont 'invite', 'question directe', 'question ouverte', 'confirmation explicite' et 'fermeture du dialogue'. Nous réalisons plusieurs expériences différentes dans lesquelles l'agent d'apprentissage évolue dans un espace d'état construit sur base de l'historique de l'interaction et d'une valeur binaire indiquant si le niveau de confiance de la dernière interaction est *haut* ou *bas*. Les expériences varient entre autre par la définition du niveau de confiance qui peut être uniquement CL_{ASR} (espace d'états S_1 dans la suite) et $CL_{ASR} * CL_{NLU}$ (espace d'états S_2 dans la suite). De même la fonction de coût intègre l'une ou l'autre mesure de confiance. Au début de chaque dialogue, un but d'utilisateur est construit assignant des valeurs aux 5 attributs. La mesure de complétion de la tâche est alors définie comme le rapport entre le nombre d'attributs dont la valeur a été correctement assignée au nombre d'attributs en tout (5 ici). On définit aussi deux environnements de simulation. Le premier (Sim_1) intègre le modèle d'utilisateur et un module de simulation de reconnaissance vocale introduisant des erreurs et une mesure de confiance de reconnaissance. Le second environnement (Sim_2) intègre, en plus, le module de compréhension. Nous avons réalisé trois expériences différentes en combinant différemment les espaces d'états et les environnements de simulation. Les résultats de l'apprentissage sont montrés dans les tableaux de la Figure 8. Dans le tableau de gauche sont indiqués les résultats des mesures objectives pouvant être obtenues lors d'un dialogue moyen suivant la stratégie apprise (mesures obtenues en calculant la moyenne des mesures faites sur 10 000 dialogues simulés). Dans le tableau de droite sont indiquées les fréquences moyennes d'occurrences de chaque type d'acte de communication.

	<i>N</i>	<i>TC</i>		invite	constQ	openQ	expC	Close
<i>Sim₁, S₁</i>	5.39	0.81	<i>Sim₁, S₁</i>	1.0	0.85	1.23	1.31	1.0
<i>Sim₂, S₁</i>	7.03	0.74	<i>Sim₂, S₁</i>	1.0	1.25	1.18	2.60	1.0
<i>Sim₂, S₂</i>	5.82	0.79	<i>Sim₂, S₂</i>	1.0	1.05	1.18	1.58	1.0

Figure 8 : Résultats de l'expérience

Grâce aux tableaux de la Figure 8, nous pouvons conclure que lors de la première expérience (sans erreur de compréhension), il y a plus de question ouvertes que de questions directes. Les erreurs de reconnaissances étant prises en compte par l'introduction de CL_{ASR} dans S_1 et Sim_1 , il y a souvent des demandes de confirmations. Dans la deuxième expérience, des erreurs de compréhensions sont introduites mais elles ne peuvent pas être détectées par les mesures de confiance. On observe une augmentation du nombre de confirmations puisque le système ne peut jamais être certain que les valeurs sont bien assignées. La longueur moyenne du dialogue s'en trouve augmentée et la complétion de la tâche diminue. En ajoutant CL_{NLU} dans S_2 , les performances s'améliorent et on retrouve presque les résultats de la première expérience. Ceci est dû au fait que certaines questions ouvertes sont évitées parce qu'elles résultent en une très mauvaise mesure de confiance. En effet la stratégie est modifiée et les questions ouvertes concernant les deux villes en même temps sont très peu probables car elles induisent des confusions et des niveaux de confiance plus faibles.

6 Conclusions et perspectives

Dans cet article, un environnement de simulation de dialogues dans lequel ont été introduit un modèle d'utilisateur consistant et un module de simulation de compréhension de parole a été décrit. Cet environnement a été développé dans le but d'un apprentissage de stratégies de dialogues optimales et il a pu être démontré par expérience que cet environnement permettait de mettre en évidence des problèmes éventuels de compréhension et d'adapter la stratégie automatiquement en conséquence. Quelques particularités de l'environnement n'ont pas été exploitées dans ce travail et il serait probablement intéressant de s'y atteler dans le futur. Par exemple, la relation avec le fonctionnement parallèle de l'utilisateur et le gestionnaire de dialogue et le phénomène de *grounding* intervenant dans les dialogues homme-homme a été brièvement mentionné dans la section 2 mais n'a pas vraiment été exploitée. Le besoin d'introduire des sous-dialogues permettant la mise en phase des connaissances supposées de l'utilisateur et de l'état réel du gestionnaire pourrait être détecté par la l'inconsistance entre l'état du système et des valeurs inférées de la connaissance de l'utilisateur.

Références

- ECKERT W., LEVIN E., PIERACCINI R. (1998) Automatic Evaluation of Spoken Dialogue Systems, *Technical Report TR98.9.1, AT&T Labs Research*.
- LEVIN E., PIERACCINI R. (1997), A Stochastic Model of Computer-Human Interaction for Learning Dialogue Strategies, *Proc. Eurospeech'97*, Rhodes, Greece, pp. 1883-1886.
- PEARL J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc. San Francisco, California.
- PIETQUIN O., DUTOIT T. (2002) Modélisation d'un Système de Reconnaissance dans le Cadre de l'Evaluation et l'Optimisation Automatique des Systèmes de Dialogue, *Actes des Journées d'Etude de la Parole, JEP 2002*, Nancy (France).
- POWER R. (1979) The Organization of Purposeful Dialogues, *Linguistics 17*, pp. 107-152.
- SINGH S., KEARNS M., LITMAN D., WALKER M., (1999) Reinforcement Learning for Spoken Dialogue Systems, *Proc. NIPS'99*, Denver, USA.