SETIT 2007
4rth International Conference: **S**ciences of **E**lectronic,
Technologies of **I**nformation and **T**elecommunications
March 25-29, 2007 – TUNISIA

Un Cadre Probabiliste pour l'Optimisation des Systèmes de Dialogue

Olivier Pietquin

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

Abstract: Dans cet article, un cadre théorique pour la simulation et l'optimisation automatique de systèmes de dialogues vocaux entre homme et machine par le biais d'un apprentissage non-supervisé de stratégies est proposé. Ce cadre s'appuie sur une description probabiliste de la communication parlée entre homme et machine. Il permet de s'inscrire dans le cadre des processus décisionnels de Markov et de faire usage de l'apprentissage par renforcement pour rechercher une stratégie optimale de manière indépendante de la tâche. Deux applications concrètes du cadre proposé aux cas du remplissage de formulaire et de l'interrogation de bases de données sont données afin d'en démontrer les utilisations possibles.

Mots Clés: Apprentissage non supervisé, Optimisation, Simulation, Système de Dialogue.
Keywords: Optimization, Simulation, Spoken Dialogue Systems, Unsupervised Learning.

INTRODUCTION

La perspective de pouvoir interagir avec des machines par le biais de la parole et du langage naturel se fait de plus en plus réaliste. Les développements de ces dernières années en matière de *reconnaissance vocale* (ASR[1]) [Boite *et al* 02] et de *compréhension du langage naturel* (NLU[2]) [Allen 94] ainsi qu'en *synthèse de parole* (TTS[3]) [Dutoit 97] rendent possible la réalisation d'interface homme-machine basée sur la parole plus souvent appelées *systèmes de dialogue vocaux* (SDS[4]). Néanmoins la conception d'interfaces vocales reste aujourd'hui une tâche relativement difficile. Il ne suffit en effet pas de juxtaposer des modules de traitement les uns aux autres pour obtenir une interface. La conception d'une stratégie d'interaction est nécessaire et même si des langages de description haut-niveau comme VoiceXML [VoiceXML] sont disponibles cela n'en reste pas moins un problème assez complexe.

Les raisons principales de cette difficulté résident dans la grande variabilité des facteurs qu'il faut prendre en compte lors du développement d'une telle interface. Ces facteurs incluent, entre autres, la nature de la tâche que le système doit réaliser (accès à une base de données, remplissage de formulaire, call routing etc.), le comportement de l'utilisateur (expérimenté, novice, coopératif etc.) et les performances des sous-systèmes tels que les systèmes ASR, NLU et TTS par exemple. La stratégie globale d'un système de dialogue devra par exemple inclure des sous-dialogues de confirmation afin de palier de mauvaises performances des systèmes de reconnaissance vocale et de compréhension de la parole.

Il apparaît donc que la conception d'une stratégie de dialogue soit une tâche d'expert à recommencer pour chaque application. De plus, la définition d'une stratégie optimale n'est pas évidente même pour un expert. C'est pourquoi, depuis le milieu des années 1990, des recherches sont menées dans le domaine de l'apprentissage non supervisé de stratégies de dialogue homme-machine. En particulier, l'application de l'*apprentissage par renforcement* (RL[5]) [Sutton & Barto 98] au problème qui nous occupe fut proposée dans [Levin *et al* 97].

L'apprentissage par renforcement est une méthode basée sur une série d'essais-erreurs qui nécessite un

[1] Automatic Speech Recognition
[2] Natural Language Understanding
[3] Text-to-Speech
[4] Spoken Dialogue Systems
[5] Reinforcement Learning

grand nombre d'exemples pour converger vers une solution optimale (de l'ordre de 10^4 dialogues pour le type d'application que nous visons). Comme il est impossible de demander à un ou plusieurs utilisateurs d'interagir réellement avec une machine autant de fois qu'il est nécessaire pour obtenir la convergence de l'algorithme pour des raisons de temps et de coût, des méthodes de simulation de dialogue vocaux ont été développées [Singh *et al* 99] [Levin *et al* 00] [Scheffler & Young 01] [López-Cózar *et al* 03] [Pietquin & Dutoit 06a].

Dans cet article, nous présentons une description probabiliste formelle de la communication parlée entre homme et machine dans le but de l'utiliser comme base pour un modèle de simulation de dialogues homme-machine. Nous étendons aussi cette description pour la rendre compatible avec le formalisme des *processus de décision de Markov* (MDP[6]) qui sont le fondement de l'apprentissage par renforcement. Cette dernière technique sera ensuite utilisée pour chercher automatiquement une stratégie optimale d'interaction. L'application de ce modèle sur des tâches d'accès à des bases de données et de remplissage de formulaire sera enfin décrite.

1. Description de la Communication Parlée entre Homme et Machine

Comme dans [Pietquin 05], nous commençons par faire une description formelle d'un dialogue vocal entre homme et machine. Celui-ci sera vu comme un cycle séquentiel durant lequel un utilisateur humain et un *gestionnaire de dialogue* (DM[7]) communiquent par la voix au travers de deux canaux composés de divers modules de traitement. Le canal transmettant la parole de l'utilisateur (modules de *traitement des entrées vocales*) est composé d'un module de *reconnaissance vocale* (ASR) et d'un module de *compréhension de langage naturel* (NLU). Le canal permettant de transformer les actes de communications abstraits générés par le gestionnaire de dialogues (modules de *génération des sorties vocales*) est, quant à lui, composé d'un module de *génération de langage naturel* (NLG[8]) [Reiter & Dale 00] et d'un système de *synthèse vocale* (TTS) comme le montre la Figure 1.

Le laps de temps entre deux interactions peut être de longueur variable et il compose un *tour*. A chaque tour t, le système de gestion de dialogue émet un ensemble d'actes de communication $\boldsymbol{a_t}$ qu'il choisit en accord avec sa stratégie interne $\boldsymbol{\pi_t}$ et en fonction de l'historique de l'interaction (représenté par la suite de ses états internes $\{\boldsymbol{s_t}\}_{t=0,\dots t}$ et de ses actes précédents $\{a_t\}_{t=0,\dots t-1}$). Un acte de communication peut être une question posée à l'utilisateur, une demande de confirmation, la présentation d'une information demandée par l'utilisateur, l'ouverture ou la fermeture du dialogue etc.

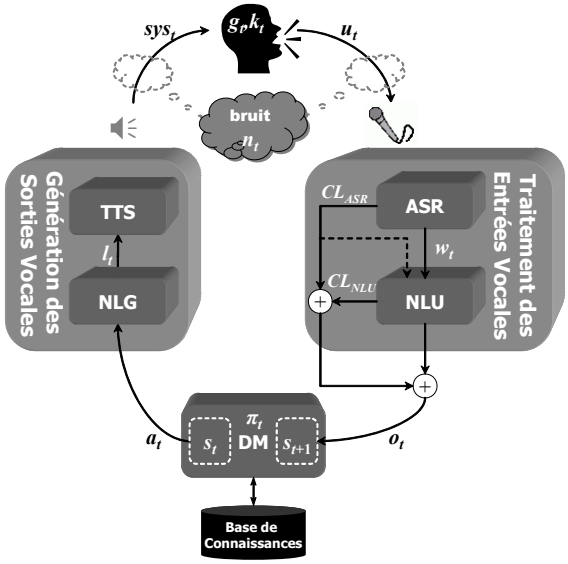

Figure 1 : Description de la communication parlée entre homme et machine

Cet acte est ensuite transformé en un ensemble de mots, un texte $\boldsymbol{l_t}$, exprimant les concepts qu'il intègre par le module de génération de langage naturel (NLG). Le texte l_t est ensuite traité par le système de synthèse de parole (TTS) pour créer le signal de parole $\boldsymbol{sys_t}$, destiné à l'utilisateur. Ce signal peut être mélangé avec un bruit ambiant $\boldsymbol{n_t}$. En fonction du sens qu'il a pu extraire de sys_t, de sa connaissance $\boldsymbol{k_t}$ à l'instant t et du but $\boldsymbol{g_t}$ qu'il poursuit en interagissant avec le système, l'utilisateur répond en produisant à son tour un signal de parole $\boldsymbol{u_t}$ qui peut être éventuellement altéré par le bruit n_t avant d'être capturé par un microphone. Ce signal est ensuite traité par le module de reconnaissance de parole (ASR) qui le transforme en une séquence de mots $\boldsymbol{w_t}$. Le module ASR produit aussi une mesure $\boldsymbol{CL_{ASR}}$ indiquant la confiance du système dans le résultat obtenu. Le système de compréhension de parole (NLU) utilise alors la séquence w_t (parfois accompagnée de la mesure CL_{ASR}) pour en extraire une signification et formulée sous forme d'une séquence de concepts $\boldsymbol{c_t}$. Similairement au module ASR, le module NLU produit une *mesure de confiance* $\boldsymbol{CL_{NLU}}$. C'est cette dernière séquence, en conjonction avec les deux mesures de confiance, qui est finalement utilisée par le module de gestion de dialogue pour former une *observation* $\boldsymbol{o_t}$ qui peut être considérée comme la réponse de son environnement à la sollicitation a_t du gestionnaire de dialogue. Celui-ci utilise alors l'observation o_t pour mettre à jour son état interne et le processus peut alors recommencer.

2. Description Probabiliste

Si nous considérons le dialogue comme décrit précédemment et comme étant le résultat du traitement par l'environnement du signal a_t émis par le gestionnaire de dialogue, nous pouvons décrire l'interaction par la probabilité suivante :

[6] Markov Decision Processes
[7] Dialogue Manager
[8] Natural Language Generation

$$P(s_{t+1},o_t,a_t \mid s_t,n_t,a_{t-1},s_{t-1},n_{t-1},\ldots,a_0,s_0,n_0) = \underbrace{P(s_{t+1} \mid o_t,a_t,s_t,n_t,a_{t-1},s_{t-1},n_{t-1},\ldots,a_0,s_0,n_0)}_{\text{Model de Tâche}} \cdot \underbrace{P(o_t \mid a_t,s_t,n_t,a_{t-1},s_{t-1},n_{t-1},\ldots,a_0,s_0,n_0)}_{\text{Environnement}} \cdot \underbrace{P(a_t \mid s_t,n_t,a_{t-1},s_{t-1},n_{t-1},\ldots,a_0,s_0,n_0)}_{\text{Gestion de Dialogue}} \quad (1)$$

Le premier terme de droite de cette égalité est relatif au *modèle de tâche* qui permet au gestionnaire de dialogue de mettre à jour son état interne en fonction de l'observation. Ces probabilités sont conditionnées par l'historique de l'interaction ce qui les rend relativement lourdes à estimer. Néanmoins, il est courant de faire l'hypothèse que, en l'absence de bruit, le choix du gestionnaire de dialogue quant aux actes de communication à générer au temps t (a_t) et l'état de celui-ci au temps t+1 (s_{t+1}) ne dépendent que de l'état précédent (s_t) et pas des états ou actions antérieurs :

$$P(s_{t+1},a_t \mid s_t,a_{t-1},s_{t-1},\ldots,a_0,s_0) = P(s_{t+1},a_t \mid s_t) \quad (2)$$

Cette hypothèse paraît évidemment très forte, néanmoins on peut relativement facilement la satisfaire en choisissant judicieusement la représentation des états. En effet, si chaque état est considéré comme renfermant suffisamment d'information pour décrire l'historique de l'interaction jusqu'à l'instant ou il a été atteint, l'hypothèse est satisfaite. Une telle représentation des états est qualifiée d'*informationnelle* et le système est alors qualifié de *Markovien*. Si, de plus, le bruit est considéré comme complètement aléatoire, les échantillons successifs de bruit $\{n_t\}$ sont alors indépendants et on peut réécrire (1) comme suit :

$$P(s_{t+1},o_t,a_t \mid s_t,n_t) = \underbrace{P(s_{t+1} \mid o_t,a_t,s_t,n_t)}_{\text{Modèle de Tâche}} \cdot \underbrace{P(o_t \mid a_t,s_t,n_t)}_{\text{Environnement}} \cdot \underbrace{P(a_t \mid s_t,n_t)}_{\text{Gestion de Dialogue}} \quad (3)$$

Le terme correspondant à l'action de l'environnement, peut alors être décomposé comme suit :

$$P(o \mid a,s,n) = \sum_{sys,k,g,u} P(o,sys,k,g,u \mid a,s,n) \quad (4)$$

Dans cette dernière égalité, les indices t ont été volontairement omis par soucis de clarté. En utilisant encore la loi des probabilités composées, on peut écrire :

$$\sum_{sys,k,g,u} P(o,sys,k,g,u \mid a,s,n) = \sum_{sys,k,g,u} \underbrace{P(sys \mid a,s,n)}_{\text{Gén. des sorties}} \cdot \underbrace{P(k \mid sys,a,s,n)}_{\text{MAJ de connaissance}} \cdot \underbrace{P(g \mid k,sys,a,s,n)}_{\text{Modification du but}} \cdot \underbrace{P(u \mid g,k,sys,a,s,n)}_{\text{Sortie utilisateur}} \cdot \underbrace{P(o \mid u,g,sys,a,s,n)}_{\text{Trait. des entrées}} \quad (5)$$

En utilisant des simplifications identiques à celle exposées dans [Pietquin 05], on obtient :

$$\sum_{sys,k,g,u} P(o,sys,k,g,u \mid a,s,n) = \sum_{sys,k,g,u} \underbrace{P(sys \mid a,s)}_{\text{Gén. des sorties}} \cdot \underbrace{P(k \mid sys,s,n)}_{\text{MAJ de connaissance}} \cdot \underbrace{P(g \mid k)}_{\text{Modification du but}} \cdot \underbrace{P(u \mid g,k,sys,n)}_{\text{Sortie utilisateur}} \cdot \underbrace{P(o \mid u,a,s,n)}_{\text{Trait. des entrées}} \quad (6)$$

2.1. Modèle d'Utilisateur

Dans (6), trois termes sont à associer au comportement de l'utilisateur :

$$\underbrace{P(k \mid sys,s,n)}_{\text{MAJ de connaissance}} \cdot \underbrace{P(g \mid k)}_{\text{Modification du but.}} \cdot \underbrace{P(u \mid g,k,sys,n)}_{\text{Sortie Utilisateur}} \quad (7)$$

Ces trois termes mettent en évidence les relations étroites qui existent entre le processus de production de parole et le couple {but, connaissance}. Une modification de la connaissance peut avoir lieu lors d'un échange et cette modification peut avoir une incidence sur le but. Le premier facteur correspond à cette éventuelle mise à jour de la connaissance de l'utilisateur. Plusieurs concepts peuvent se cacher derrière le terme de connaissance et dans le cas particulier des systèmes de dialogue, ce terme peut faire référence à la connaissance de l'utilisateur en ce qui concerne l'*historique* de l'interaction, la *tâche*, le *système* lui-même ou le *monde* en général. Cette connaissance n'est modifiable que par un processus incrémental et ne peut être complètement remise à zéro à chaque tour. Dans ces conditions, on peut écrire :

$$P(k \mid sys,s,n) = \sum_{k^-} P(k \mid k^-,sys,s,n) \cdot P(k^- \mid sys,s,n) = \sum_{k^-} P(k \mid k^-,sys,n) \cdot P(k^- \mid s), \quad (8)$$

où $k^- = k_{t-1}$. La simplification du second terme 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. De (8), il peut être conclu que, tout comme le fonctionnement du gestionnaire de dialogue qui est basé sur la mise à jour de son état interne grâce à l'observation o, le comportement de l'utilisateur repose sur une mise à jour de sa connaissance grâce au signal de parole *sys*. La mise à jour ne s'opérant pas sur le même espace d'états ceci est une source possible d'erreurs de compréhension entre les deux protagonistes. Dans une certaine mesure, on peut comparer ceci avec le phénomène de « *grounding* » intervenant dans un dialogue homme-homme [Clark & Shaeffer 89].

2.2. Traitement des Entrées Vocales

Le traitement des entrées vocales est représenté par le terme $P(o \mid u, a, s, n)$. Celui-ci peut encore être factorisé comme suit :

$$
\begin{aligned}
P(o \mid u,a,s,n) &= P(c, CL_{ASR}, CL_{NLU} \mid u,a,s,n) \\
&= \sum_w P(c, CL_{ASR}, CL_{NLU} \mid w,u,a,s,n) \cdot P(w \mid u,a,s,n) \\
&= \sum_w P(c, CL_{NLU} \mid w, CL_{ASR}, u,a,s,n) \cdot P(w, CL_{ASR} \mid u,a,s,n) \\
&= \sum_w P(c, CL_{NLU} \mid w, CL_{ASR}, a,s) \cdot P(w, CL_{ASR} \mid u,a,s,n) \\
&\approx \max_w \underbrace{P(c, CL_{NLU} \mid w, CL_{ASR}, a,s)}_{\text{NLU}} \cdot \underbrace{P(w, CL_{ASR} \mid u,a,s,n)}_{\text{ASR}}.
\end{aligned} \tag{9}
$$

Dans cette dernière équation, la somme peut souvent être remplacée par le terme maximum intervenant dans celle-ci du fait de la nature du processus de reconnaissance vocale qui consiste à déterminer la séquence *w* qui permet de maximiser la probabilité $P(w, CL_{ASR} \mid u,a,s,n)$. Le facteur associé à la compréhension de parole a pu être simplifié puisque le processus ne prend en compte que les séquences de mots *w* résultant de la reconnaissance vocale et pas le véritable signal de parole *u* prononcé par l'utilisateur ni le bruit *n* pouvant affecter le résultat de reconnaissance vocale. Le terme associé à la reconnaissance vocale quant à lui ne peut plus être simplifié. Néanmoins, il peut être décomposé comme suit :

$$
\begin{aligned}
P(w, CL_{ASR} \mid u,a,s,n) = P(CL_{ASR} \mid w,u,a,s,n) \cdot \\
P(w \mid u,a,s,n)
\end{aligned} \tag{10}
$$

La génération du niveau de confiance de reconnaissance vocale étant généralement basée uniquement sur les phénomènes acoustiques, les variables *a* et *s* conditionnent rarement le premier facteur du produit ci-dessus. Grâce à la loi de Bayes, le second facteur peut être encore transformé pour permettre de retrouver l'équation habituelle à la base de la plupart des algorithmes de reconnaissance de formes permettant de calculer la probabilité *a posteriori* grâce à la probabilité *a priori* :

$$
P(w \mid u,a,s,n) = \frac{P(u \mid w,a,s,n) \cdot \overbrace{P(w \mid a,s,n)}^{\text{Modèle de Langage}}}{P(u \mid a,s,n)} \tag{11}
$$

Le terme $P(w \mid a, s, n)$ représente le modèle de langage (c'est à dire la probabilité d'occurrence d'un mot dans le langage étudié). Ce terme peut aussi être simplifié puisqu'il est indépendant du bruit. Par contre, il reste conditionné par *a* et *s* et c'est une caractéristique importante des systèmes de dialogue que d'être capable d'adapter le modèle de langage en fonction du contexte. En effet, afin d'améliorer les performances de reconnaissance, un système de dialogue peut n'accepter que les phrases répondant à une question posée comme entrées valides et donc limiter le nombre de possibilités. Comme d'après l'expression (9), seule la maximisation sur *w* est intéressante, le dénominateur de (11) qui est indépendant de *w* peut disparaître. Dans ces conditions, le résultat $\boldsymbol{w_h}$ du processus de reconnaissance vocale est obtenu sur base de l'équation suivante :

$$
w_h = \arg\max_w P(u \mid w,a,s,n) \cdot P(w \mid a,s,n) \tag{12}
$$

De manière similaire, le terme associé au processus de compréhension de parole peut être décomposé comme suit :

$$
\begin{aligned}
P(c, CL_{NLU} \mid w, CL_{ASR}, a,s) = P(c \mid w, CL_{ASR}, a,s) \cdot \\
P(CL_{NLU} \mid c, w, CL_{ASR}, a,s)
\end{aligned} \tag{13}
$$

Il s'agit alors pour le module chargé de la compréhension de parole de maximiser la probabilité d'une séquence de concepts *c* étant donnée la séquence de mots *w*. Le résultat $\boldsymbol{c_h}$ doit donc satisfaire :

$$
c_h = \arg\max_c P(c \mid w, CL_{ASR}, a,s) \tag{14}
$$

Cette dernière égalité est à la base de la plupart des systèmes stochastiques de compréhension du langage naturel [Pieraccini & Levin 92].

2.3. Génération de Parole

Enfin, le terme de l'égalité (6) associé à la génération de parole peut lui aussi être décomposé de la manière suivante :

$$
\begin{aligned}
P(sys \mid a,s) &= \sum_l P(sys, l \mid a,s) \\
&= \sum_l P(sys \mid l,a,s) \cdot P(l \mid a,s) \\
&= \sum_l \underbrace{P(sys \mid l)}_{\text{TTS}} \cdot \underbrace{P(l \mid a,s)}_{\text{NLG}}
\end{aligned} \tag{15}
$$

Etant donnée la dépendance exclusive du processus de synthèse de parole au texte à synthétiser, le terme associé (TTS) a été simplifié. La plupart des systèmes de dialogue n'utilisent pas réellement de génération de langage naturel, c'est à dire de synthèse automatique de texte à partir de concepts [Reiter & Dale 00]. De manière générale, les textes sont rédigés par les concepteurs du système pour être ensuite transformés en parole synthétique (la synthèse vocale est parfois même inexistante et le système utilise des enregistrements de voix naturelle). Ces dernières années néanmoins, le développement de systèmes NLG dédiés aux systèmes de dialogue fait l'objet de recherches spécifiques [Walker *et al* 02] et le résultat n'est donc pas toujours déterministe. Ceci justifie donc la description probabiliste utilisée pour ce processus. Le processus de génération de texte est dépendant de l'état dans lequel se trouve le système car, suivant l'historique de l'interaction, on pourra par exemple pronominaliser certains sujets ou compléments s'ils ont déjà été mentionnés plus tôt dans le dialogue ou réaliser des anaphores.

3. Optimisation

A priori, personne ne peut décrire la stratégie de dialogue idéal, même en connaissant parfaitement l'application à laquelle est dédié le système. Celle-ci dépend en effet d'un grand nombre de facteurs, tels que les performances des modules composant le systèmes de dialogue, l'expertise de l'utilisateur dans

le domaine etc. C'est pourquoi l'apprentissage supervisé de stratégies de dialogue est impossible. Dans cette optique, l'utilisation d'algorithmes d'apprentissage non supervisé comme l'apprentissage par renforcement fut proposée pour optimiser les systèmes de dialogue [Levin *et al* 97].

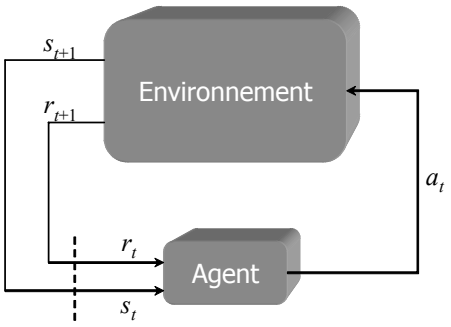

Figure 2 : Apprentissage par renforcement

Pour ce faire, il est nécessaire de décrire un dialogue homme-machine comme un *processus de décision de Markov* (MDP). Un tel processus, comme le montre la Figure 2, fait intervenir un système d'apprentissage appelé *agent* qui interagit séquentiellement aux instants *t* avec un *environnement* par le biais d'*actions* $\{a_t\}$. Après chaque action, l'agent observe une modification de l'*état* s_{t+1} de l'environnement et un *coût* r_{t+1} associé. Le but de l'agent est alors d'apprendre une stratégie optimale $\boldsymbol{\pi^*}$, c'est-à-dire une correspondance entre état observé et action qui lui permette de minimiser le coût total de son interaction sur le long terme (et donc pas forcément le coût de chaque action). Ce coût total $\boldsymbol{R_t}$ de l'interaction à partir du temps *t* est défini comme une somme pondérée des coûts instantanés $\{r_i\}_{i=t+1\to\infty}$ généralement de la forme :

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (16)$$

La stratégie optimale est alors définie par :

$$\pi^* = P_{\pi^*}(a \mid s) = \arg\min_{\pi} E_{\pi}[R_t] \quad (17)$$

On suppose enfin que le processus répond à la propriété de Markov, c'est-à-dire que l'état et le coût au temps *t*+1 dépendent uniquement de l'état et de l'action au temps *t* :

$$P(s_{t+1}, r_{t+1} \mid s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_1, a_1) = P(s_{t+1}, r_{t+1} \mid s_t, a_t) \quad (18)$$

Le processus de décision de Markov est alors complètement défini par l'espace des états $\boldsymbol{S} = \{s_i\}$, l'ensemble des actions possibles pour l'agent $\boldsymbol{A} = \{a_j\}$ et par la dynamique à un pas de l'environnement donnée par la probabilité de transition entre états $\mathcal{T}$ et la distribution des coûts $\mathcal{R}$:

$$\begin{aligned} \mathcal{T}_{ss'}^{a} &= P(s_{t+1} = s' \mid s_t = s, a_t = a) \\ \mathcal{R}_{ss'}^{a} &= E[r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'] \end{aligned} \quad (19)$$

On comprend rapidement que ce qu'il manque pour définir un MDP en se basant sur le modèle de dialogue décrit dans le début de cet article, c'est la définition d'un coût instantané associé à la réponse *o* de l'environnement quand il est sollicité par l'acte *a* du gestionnaire de dialogue. En effet, en comparant (3) et (19) on peut écrire :

$$\mathcal{T}_{ss'}^{a} = \underbrace{P(s_{t+1} \mid o_t, a_t, s_t)}_{\text{Modèle de Tâche}} \cdot \underbrace{P(o_t \mid a_t, s_t)}_{\text{Environnement}} \quad (20)$$

On peut écrire (19) sans perte de généralité en incorporant le bruit dans la définition de l'état. Les développements qui ont suivit (3) permettent de décomposer $\mathcal{T}$ comme suit :

$$\begin{aligned} \mathcal{T}_{ss'}^{a} \approx \max_{w} \sum_{sys,l,k,k^-,g,u} & P(sys \mid l) \cdot P(l \mid a, s) \cdot \\ & P(k \mid k^-, sys, n) \cdot P(k^- \mid s) \cdot P(g \mid k) \cdot P(u \mid g, k, sys, n) \cdot \\ & P(u \mid w, a, s, n) \cdot P(w \mid a, s, n) \cdot P(CL_{ASR} \mid w, u, a, s, n) \\ & P(c \mid w, CL_{ASR}, a, s) \cdot P(CL_{NLU} \mid c, w, CL_{ASR}, a, s) \cdot \\ & P(s' \mid o, a, s) \end{aligned} \quad (21)$$

Afin de définir un coût, il a été proposé dans [Singh *et al* 99]] d'utiliser la contribution de la réalisation d'une action à la satisfaction de l'utilisateur. Cette notion étant relativement difficile à utiliser dans un système automatique de part sa subjectivité, l'étude décrite dans [Walker *et al* 970] propose d'estimer cette notion en utilisant les facteurs objectifs contribuant à la satisfaction de l'utilisateur, à savoir les performances de reconnaissance et de compréhension du système de dialogue, la durée en temps d'un dialogue et le degré de complétion de la tâche à la fin du dialogue. Le modèle statistique de l'environnement défini plus tôt permet d'évaluer les performances de reconnaissance et de compréhension du système par le biais des variables CL_{ASR} et CL_{NLU} et la durée en temps du dialogue peut être estimée par le nombre de tours nécessaires pour terminer un dialogue. Malheureusement, le degré de complétion de la tâche est une donnée trop dépendante du type de tâche étudié et devra être ajusté au coup par coup. Ainsi, nous aurons une fonction de coût de la forme :

$$r_{t+1} = w_N \cdot N_t - w_{CL} \cdot f(CL_{ASR,t}, CL_{NLU,t}) - w_{TC} \cdot TC_t \quad (22)$$

où les w_x sont des poids positifs ajustables, *N* est égal à 1 si le tour considéré n'est pas le dernier et 0 sinon (moyen de compter les tours) et *TC* est une mesure du degré de complétion de la tâche qui restera à définir selon la tâche considérée.

4. Construction d'un Modèle Pratique

La description théorique d'un dialogue homme-machine faite précédemment a permis d'isoler formellement les paramètres intervenant dans un tel processus, de mettre en évidence les interactions entre ces paramètres et d'inscrire le dialogue homme-machine dans le cadre des processus de décision de Markov. Néanmoins, l'utilisation pratique de cette description dans l'optique d'une optimisation de la stratégie nécessite d'estimer les paramètres du MDP et particulièrement la dynamique à un pas $\{\mathcal{T}, \mathcal{R}\}$. Plutôt que d'estimer ces paramètres et d'essayer de résoudre le problème de minimisation du coût global

analytiquement, nous nous proposons de construire un environnement virtuel produisant par simulation des transitions et des coûts, basé sur la description probabiliste de l'environnement, et d'apprendre en ligne la stratégie optimale par interaction directe comme proposé dans [Levin *et al* 97].

Pour ce faire nous avons utilisé un *réseau bayésien dynamique* (DBN[9]) [Pearl 88] pour obtenir les paramètres correspondant au comportement de l'utilisateur comme décrit dans [Pietquin & Dutoit 06b]. Ce modèle a aussi été utilisé pour simuler le module de compréhension de parole, toujours selon la méthode décrite dans [Pietquin & Dutoit 06b]. La modélisation du système de reconnaissance de parole a été faite par classification des tâches de reconnaissance et par paramètrisation de ces tâches d'après des données réelles (en termes de taux de reconnaissance et de distribution des niveaux de confiance) suivant la méthode décrite dans [Pietquin & Renals 02]. D'autres méthodes de simulation du système de reconnaissance vocale peuvent être envisagées comme celle décrites dans [Pietquin & Beaufort 05]

5. Exemple d'utilisation

Notre modèle théorique a été testé pour l'apprentissage de stratégies de dialogue dans le cadre de deux applications : le remplissage de formulaire et l'accès aux bases de données. Ces exemples sont des tâches simplifiées permettant de mettre en évidence certaines qualités du système. Pour ce faire, pour chacune des tâches, il est nécessaire de définir les types d'actes de communication autorisés pour le système, la fonction des niveaux de confiance à intégrer dans la fonction de coût ainsi que la mesure de complétion de la tâche et la définition de l'espace des états.

5.1. Accès Vocal à une Base de Données

Ici, la tâche à optimiser est la consultation vocale d'une base de données. Dans ce cadre, une base de données contenant les caractéristiques de 350 ordinateurs fut utilisée. Cette base est divisée en 2 tables (une pour les ordinateurs de bureau, l'autre pour les ordinateurs portables) chacune contenant 6 champs : pc ou mac, type de processeur (Pentium, AMD etc.), cadence (MHz), taille mémoire (Mo), taille du disque dur (Mo), marque. Le but du système de dialogue est alors d'obtenir de l'utilisateur suffisamment d'information pour extraire un nombre limité d'enregistrements intéressants de la base. Pour ce faire, le système (et donc l'agent d'apprentissage) dispose de 7 types d'acte de communication : invite, question contraignante (demandant la valeur d'un champ ou d'une table), confirmation directe (demande de confirmation d'une seule valeur), confirmation de toutes les valeurs, relaxation d'une contrainte (utile pour éviter de ne pas trouver d'enregistrement correspondant à la requête), requête à la base de données, fermeture du dialogue. Nous avons défini le but de l'utilisateur *g* comme étant un enregistrement de la base (le but est donc toujours atteignable) et la mesure de complétion de la tâche *TC* entrant dans la fonction de coût (22) est alors définie comme le ratio moyen du nombre de valeurs communes entre les enregistrements retrouvés par le système dans la base et celles contenues dans le but de l'utilisateur. La fonction du niveau de confiance intervenant dans (22) est $CL_{ASR} * CL_{NLU}$. Il reste alors à définir l'espace des états. Ici, chaque état est représenté comme un vecteur de 9 valeurs binaires décomposé comme suit : 1 valeur indiquant si la table est connue (l'utilisateur veut-il un ordinateur portable ou de bureau), 6 valeurs correspondant à chacun des champs indiquant si la valeur associée est connue, 1 valeur indiquant si le niveau de confiance est *haut* ou *bas* (nous avons défini un seuil), une valeur indiquant si le nombre d'enregistrements extraits de la base lors de la dernière requête est *élevé* ou *faible*. Cet espace d'états permet à l'agent d'apprentissage d'en savoir suffisamment sur l'historique de l'interaction que pour supposer le processus Markovien.

En utilisant les paramètres décrits ci-dessus dans un processus d'apprentissage par renforcement de type Q-learning [Sutton & Barto 98], l'agent d'apprentissage tend vers la définition de la stratégie suivante :

1. soumettre l'invite à l'utilisateur
2. si, à l'invite, l'utilisateur a fourni au moins 3 valeurs sur 7 (tables ou champs), passer à l'étape 4
3. poser des questions contraignantes sur les valeurs jusqu'à en obtenir suffisamment en suivant un ordre précis maximisant l'espérance du niveau de confiance obtenu, c'est à dire en s'intéressant d'abord aux valeurs binaires (PC ou Mac), puis aux valeurs chiffrées (cadence, taille mémoire …) et pratiquement jamais aux marques car elles fournissent des performances de reconnaissance assez mauvaises.
4. effectuer la requête à la base de données grâce aux valeurs récupérées
5. si le nombre d'enregistrements est non nul et faible passer à l'étape 8
6. si le nombre d'enregistrements est nul, demander la relaxation de certaines contraintes
7. si le nombre d'enregistrements est encore trop élevé, retourner à l'étape 3
8. présenter les résultats à l'utilisateur et fermer le dialogue.

Ce qui est évidemment intéressant ici, outre le fait qu'une stratégie cohérente a été apprise automatiquement, c'est de noter l'importance de l'introduction du niveau de confiance qui permet à l'étape 3 de suivre une séquence de questions contraignantes particulière automatiquement dérivée des problèmes pouvant intervenir dans le processus de

[9] Dynamic Bayesian Network

reconnaissance vocale.

5.2. Remplissage de Formulaire

Dans ce cas, la tâche à optimiser consiste à remplir un formulaire dans le cadre de la vente d'un ticket de train. Il ne s'agit pas ici du processus de vente lui même mais bien de recueillir les informations relatives au voyage envisagé c'est à dire la ville de départ, la ville d'arrivée, la date du départ, l'heure du départ, l'heure d'arrivée et la classe (donc 6 valeurs). Le choix des villes se fait parmi un ensemble de 50 noms, cet ensemble est commun pour les villes d'arrivée et de départ. Les heures d'arrivée et de départ possibles font aussi partie d'un ensemble commun.

Dans le cadre de cette expérimentation, les actes de communications possibles sont réduits au nombre de 5 : invite, question contraignante, question ouverte (portant sur 2 ou 3 informations en même temps), confirmation directe et fermeture du dialogue. Avant chaque dialogue, le but de l'utilisateur sera initialisé avec au plus 6 valeurs aléatoires (avec des valeurs différentes pour les villes et les heures) et la mesure de la complétion de la tâche sera le nombre de valeurs communes entre le but de départ et l'information recueillie par le système à la fin du dialogue. La fonction du niveau de confiance à placer dans l'expression (22) est la même que précédemment ($CL_{ASR} * CL_{NLU}$). L'espace des états sera construit de la même manière que précédemment en supprimant la variable binaire associée au nombre d'enregistrements retourné par la dernière requête à la base de données puisque aucune base de données n'est considérée ici.

En utilisant les paramètres décrits ci-dessus, toujours dans le cadre d'un processus Q-Learning, l'agent tend vers la définition de la stratégie suivante :

1. soumettre l'invite à l'utilisateur
2. si toutes les variables binaires associées aux valeurs à demander à l'utilisateurs sont positives et que le niveau de confiance est *haut* passer à l'étape 6
3. poser des questions ouvertes sur 2 valeurs manquantes maximum en mélangeant seulement ville et heure, ville et classe, heure et classe
4. si le niveau de confiance est *bas* demander confirmation explicite des dernières valeurs reçues
5. retour à l'étape 2
6. afficher le résultat à l'utilisateur et fermer le dialogue.

Dans cet exemple, ce qu'il est intéressant de noter, c'est l'option prise à l'étape 3. En effet, l'agent a appris à poser des questions ouvertes permettant de minimiser le temps de l'interaction (puisque les questions ouvertes permettent de récupérer plusieurs données à la fois) mais en ne demandant pas les villes d'arrivée et de départ en même temps ou les heure d'arrivée et de départ en même temps afin de minimiser le risque d'obtenir un mauvais niveau de confiance en matière de compréhension de parole. En effet, les villes de départ et d'arrivée faisant partie d'un même ensemble de valeurs, leur affectation à une ou l'autre catégorie est difficile. Il en est de même pour les heures.

6. Conclusion et Perspectives

Dans la première partie de cet article, nous avons de donné une représentation formelle de la communication parlée entre un utilisateur humain et un système de dialogue vocal. Cette description formelle mis en évidence les paramètres entrant en considération lors d'une telle communication et l'ensemble des interactions possibles entre ces paramètres. Le cadre probabiliste ainsi défini a permis d'inscrire le dialogue homme-machine dans le formalisme des processus décisionnels de Markov en donnant une expression analytique à la fonction de transition entre états et en modélisant une fonction de coût basée sur l'évaluation de l'apport de chaque acte de communication à la satisfaction de l'utilisateur.

Une seconde partie de cet article a été consacrée à l'utilisation pratique du formalisme défini plus tôt dans le cadre de l'apprentissage automatique de stratégies de dialogue. Dans cette optique, deux applications simples ont été décrites : l'accès vocal à une base de données et le remplissage de formulaire. Un algorithme d'apprentissage par renforcement a alors été utilisé pour apprendre par interaction directe une stratégie optimale pour ces deux applications. Il a été mis en évidence dans ces exemples que le système apprend réellement une stratégie possible à mettre en œuvre et que cette stratégie s'adapte particulièrement aux difficultés de la tâche étudiée.

Dans le futur, une des perspectives intéressante serait probablement d'approfondir la similarité entre l'aspect incrémental du modèle d'utilisateur (mise à jour de la connaissance) avec le phénomène de *grounding* mentionnée dans la partie consacrée à ce modèle. Par exemple, des nouveaux types d'acte de communication permettant de remettre à égalité les connaissances de l'utilisateur et du gestionnaire de dialogue pourraient alors être introduit et une amélioration des performances du système serait certainement envisageable.

REFERENCES

[Allen 94] Allen, J. *Natural Language Understanding*. Benjamin Cummings, 1987, Second Edition, 1994.

[Boite *et al* 00] Boite, R., Bourlard, H., Dutoit, T., Hancq, J., Leich H. *Traitement de la Parole*, 2nd Edition. Presses Polytechniques Universitaires Romandes, Lausanne, ISBN 2-88074-388-5, 2000.

[Clark & Schaefer 89] Clark, H., Schaefer E. Contributing to Discourse, *Cognitive Science, 13*, 1989, pp.259-294.

[Dutoit 97] Dutoit, T. *An Introduction to Text-To-Speech Synthesis*, Kluwer Academic Publishers, Dordrecht, 320 pp., ISBN 0-7923-4498-7, 1997.

[Levin *et al* 97] Levin, E., Pieraccini, R., Eckert, W., Learning Dialogue Strategies within the Markov

Decision Process Framework. *Proc. ASRU'97*, Santa Barbara, California, 1997.

[Levin *et al* 00] Levin, E., Pieraccini, R., Eckert, W. A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies, *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 11-23, 2000.

[López-Cózar *et al* 03] López-Cózar, R., de la Torre, A, Segura, J., Rubio, A. Assesment of Dialogue Systems by Means of a New Simulation Technique. *Speech Communication*, vol. 40, 2003.

[Pearl 88] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc. San Francisco, California, 1988.

[Pieraccini & Levin 92] Pieraccini, R., Levin, E. Stochastic Representation of Semantic Structure for Speech Understanding, *Speech Communication, vol. 11*, 1992, pp. 238-288.

[Pietquin & Renals 02] Pietquin, O., Renals, S. ASR System Modeling for Automatic Evaluation and Optimization of Dialogue Systems.' *Proc. ICASSP'02*, Orlando, 2002

[Pietquin 05] Pietquin, O. A Probabilistic Description of Man-Machine Spoken Communication. *Proc. ICME'05*, Amsterdam, The Netherlands, 2005.

[Pietquin & Beaufort 05] Pietquin, O., Beaufort, R. Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning. *Proc. of Eurospeech'05*, Lisbon, Portugal, 2005.

[Pietquin & Dutoit 06a] Pietquin, O., Dutoit, T. A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. *IEEE Transactions on Audio, Speech and Language Processing, Volume 14, Issue 2,* March 2006, pp 589-599.

[Pietquin & Dutoit 06b] Pietquin, O., Dutoit, T. Dynamic Bayesian Networks for NLU Simulation with Applications to Dialog Optimal Strategy Learning, *Proc. ICASSP'06*, Toulouse, France, 2006

[Reiter & Dale 00] Reiter, E., Dale, R. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, 2000.

[Scheffler & Young 01] Scheffler, K., Young, S. Corpus-Based Dialogue Simulation for Automatic Strategy Learning and Evaluation. *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, 2001.

[Singh *et al* 99] Singh, S., Kearns, M., Litman, D., Walker, M., Reinforcement Learning for Spoken Dialogue Systems. *In Proceedings of NIPS'99*, Denver, USA, 1999.

[Sutton & Barto 98] Sutton, R. and Barto, A. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[VoiceXML] VoiceXML Forum, http://www.voicexml.org/

[Walker *et al* 02] Walker, M., Rambow, O., Rogati, M. 'Training a Sentence Planner for Spoken Dialogue Using Boosting.' *Computer Speech and Language Special Issue on Spoken Language Generation* , July 2002

[Walker *et al* 97] Walker, M., Litman, D., Kamm, C., Abella, A. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. *Proc. 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, 1997, pp. 271-280.