

Modélisation d'un Système de Reconnaissance pour l'Apprentissage Automatique de Stratégies de Dialogue Optimales

Olivier Pietquin Thierry Dutoit

Faculté Polytechnique de Mons
Service de Théorie des Circuits et de Traitement du Signal
1 Avenue Copernic (Parc Initialis) – 7000 Mons – BELGIQUE
Tél.: ++32 (0)65 37 47 81 - Fax: ++32 (0)65 37 47 29
Mél: {pietquin,dutoit}@tcts.fpms.ac.be - http://tcts.fpms.ac.be/~{pietquin,dutoit}

ABSTRACT

This last decade, the field of spoken dialogue systems has developed quickly. However, rapid design of dialogue strategies remains uneasy. Automatic strategy learning has been investigated and the use of Reinforcement Learning algorithms introduced by Levin and Pieraccini is now part of the state of the art in this area. Obviously, the learned strategy's worth depends on the definition of the optimization criterion used by the learning agent and on the exactness of the environment model.

In this paper, we propose to introduce a model of an ASR system in the simulated environment in order to enhance the learned strategy. To do so, we brought recognition error rates and confidence levels produced by ASR systems in the optimization criterion.

1. INTRODUCTION

Ces dernières années, les recherches dans le domaine des systèmes de dialogue ont connu une expansion majeure et l'apprentissage automatique des stratégies de dialogue fait partie des champs d'investigation les plus importants. Dans ce cadre, la théorisation des systèmes de dialogue selon le formalisme des "Processus de Décision de Markov" (MDPs) et l'application de l'apprentissage par renforcement (Reinforcement Learning : RL) a été proposée par Pieraccini et Levin [1]. Ce type d'apprentissage non supervisé nécessite soit une série d'interactions réelles entre le système d'apprentissage (appelé agent) et un utilisateur humain au travers d'un système de reconnaissance vocale (ASR), soit une quantité importante de données issues de corpus de dialogue obtenus par des techniques de "Wizard of Oz", soit une série d'interactions entre l'agent et un utilisateur virtuel [2]. Cette dernière solution est souvent préférée puisque plusieurs milliers de dialogues peuvent être nécessaires pour entraîner un tel système.

Pour envisager l'apprentissage automatique des stratégies de dialogue dans le cadre des MDPs en utilisant les algorithmes du RL, il est évidemment nécessaire d'exprimer le design d'un système de dialogue comme un problème d'optimisation et donc de définir la notion de

coût de dialogue qu'il conviendra de minimiser lors de l'apprentissage.

Dans le but de s'approcher au plus près des conditions réelles dans lesquelles un utilisateur interagit avec le système de dialogue grâce à un dispositif ASR imparfait, nous avons intégré un modèle de système ASR dans l'environnement de simulation. De cette manière, nous avons pu introduire les taux d'erreur et les niveaux de confiance de reconnaissance vocale dans le critère d'optimisation utilisé lors de l'apprentissage par l'agent.

Les expériences dans cet environnement simulé indiquent que l'introduction des niveaux de confiance dans le système d'apprentissage apporte une amélioration notable dans les stratégies effectivement apprises. Les résultats pourraient aussi être utilisés pour l'évaluation objective des coûts de dialogue afin de pouvoir comparer les stratégies entre elles.

2. LES DIALOGUES DANS LE CADRE DES MDPs

Il est possible d'exprimer un système de dialogue en termes d'états, d'actions et de stratégie ce qui permet de les formaliser comme des Processus de Décision de Markov (MDP). Pour cela, la propriété de Markov doit être satisfaite c'est à dire que l'état s_{t+1} du système au temps $t+1$ ne doit dépendre que de l'état s_t au temps t et de l'action a_t prise par le système dans l'état s_t .

$$P(s_{t+1} | s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0) = P_T(s_{t+1} | s_t, a_t)$$

où P_T est la probabilité de transition.

Le formalisme des MDP peut être décrit comme suit :

- s_t est l'état du système au temps t . Sa représentation est construite de telle manière qu'elle décrit l'information obtenue par le système jusqu'au temps t . Pour contourner les contraintes imposées par la propriété de Markov, elle peut aussi comprendre une information sur l'historique du dialogue. Au temps t_0 l'état du système est s_0 et l'état final atteint au temps T_F est l'état particulier S_F .

- a_t est l'action accomplie au temps t par le système. Les actions font partie d'un ensemble fini d'actions $\underline{A} = \{a_i\}$. Une action est généralement une phrase parlée ou une requête de base de données.

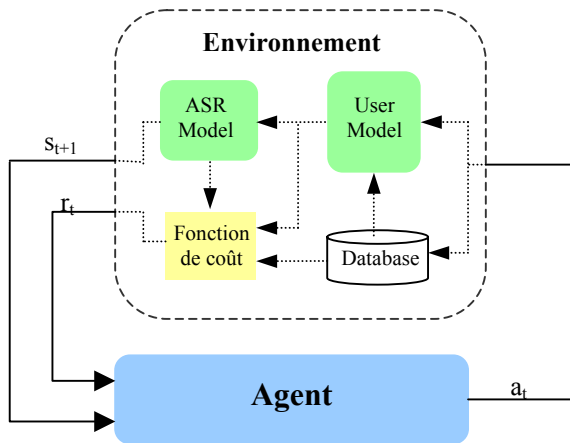


Figure 1 : Processus d'apprentissage

- r_t est le signal de renforcement ou le coût partiel associé au fait d'avoir accompli l'action a_t dans l'état s_t . Le coût d'un dialogue peut alors être exprimé comme la somme des coûts partiels obtenus durant celui-ci.

$$R_d = \sum_{t=0}^{t=T_f} r_t$$

- π est la stratégie du système. Elle régit le comportement du système en établissant une correspondance entre l'espace des états $\mathcal{S} = \{s_i\}$ et l'ensemble des actions \mathcal{A} . La fonction $\pi(s_i)$ définit l'action a_i à accomplir dans l'état s_i . La stratégie π^* est optimale si elle minimise l'espérance du coût d'un dialogue.

$$\overline{R_d} = E[R_d]$$

3. SIMULATION DE L'ENVIRONNEMENT

L'entraînement d'un système de ce type peut demander plusieurs milliers de dialogues. Pour éviter les tests sur des utilisateurs réels qui s'avèreraient extrêmement ennuyeux et contraignants, nous avons créé un utilisateur virtuel comme Pieraccini, Levin et Eckert le proposent dans [2]. Néanmoins, nous avons été un peu plus loin dans la modélisation de l'environnement en y introduisant un modèle de système ASR imparfait (figure 1). Un bloc est aussi dédié à la construction de la fonction de coût. Enfin, les systèmes de dialogue actuels étant principalement dédiés à la consultation vocale de bases de données, une telle base est donc présente dans l'environnement.

3.1. Communication conceptuelle

Les communications au sein de l'environnement de simulation et avec l'agent sont réalisées au niveau conceptuel ou au niveau de l'intention. Un concept peut être défini comme l'unité minimale d'information qu'un des participants au dialogue peut communiquer.

La modélisation de l'environnement à un niveau inférieur (comme le niveau des mots ou du signal sonore) n'est évidemment pas intéressante puisque la notion de

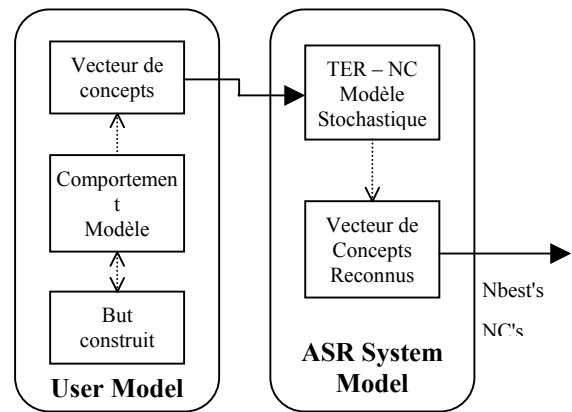


Figure 2 : Utilisateur virtuel et modèle de système ASR. (TER : Taux d'Erreur de Reconnaissance, NC : Niveau de Confiance)

stratégie est une notion de haut niveau et qu'il est, de toutes manières, plus aisé de générer des concepts que des suites de mots ayant un sens. De plus, ceci permet de simuler toutes les parties de l'environnement, y compris les systèmes de traitement automatique du langage naturel (TALN).

3.2. Utilisateur Virtuel

Pour atteindre une ergonomie optimale, il faut pouvoir offrir la possibilité à l'utilisateur de prendre l'initiative du dialogue (initiative mixte). De plus, un utilisateur réel possède un but précis lorsqu'il utilise un système de dialogue (goal-directed task). Ces observations nous ont conduits à réaliser un utilisateur virtuel (Figure 2) autorisant un comportement mixte et possédant un but construit aléatoirement à chaque début de session.

Sur base de ce but, construit d'après la base de données, l'utilisateur virtuel produit des vecteurs de concepts pour répondre aux questions de l'agent. Une partie du modèle est stochastique et permet un comportement mixte car elle contient des probabilités de répondre à des questions multiples ou à des questions non-posées ainsi que des probabilités de relaxer des contraintes et de confirmer ou infirmer des propositions de l'agent. L'utilisateur virtuel peut aussi mettre fin au dialogue prématurément et exprimer son mécontentement si le dialogue ne l'a pas satisfait.

Le but construit en début de session peut aussi servir pour donner une estimation de la satisfaction de l'utilisateur suivant qu'il ait été atteint ou non.

3.3. Modèle du système de reconnaissance

Bien que la littérature compte quelques descriptions de systèmes d'apprentissage par renforcement des stratégies et d'environnements de simulation de dialogue comme [1] et [3], peu d'entre eux intègrent une modélisation d'un système de reconnaissance vocale. Si c'est le cas, la modélisation se limite à un simple *taux d'erreurs de*

reconnaissance (TER) et il n'est jamais tenu compte des spécificités des tâches de reconnaissance.

Dans le but d'améliorer les stratégies apprises, nous avons implémenté un modèle plus complexe de système ASR (Figure 2) incluant plusieurs distributions de TER et de niveaux de confiance (NC), comme c'est le cas dans la réalité. Le NC peut être défini comme un nombre réel compris en 0 et 1 et qui donne une estimation de la confiance du système dans le résultat de reconnaissance qu'il produit. Il est calculé sur base du signal acoustique de départ et sa distribution est décrite par deux courbes distinctes correspondant respectivement aux mots bien et mal reconnus [4]. La figure 3 montre une telle distribution obtenue sur un système ASR réel mais en utilisant des données extraites de ses données d'entraînement. Les courbes devraient donc être un peu plus aplanies en réalité. Néanmoins, l'apprentissage étant basé sur la relativité des performances de reconnaissances, les résultats absolus ne sont pas très importants dans le cas qui nous occupe.

Les performances d'un système ASR ne sont pas égales suivant qu'il s'agisse de reconnaître un booléen, une suite de chiffres, une suite de nombres, une date, une suite de mots distincts ou un flot de parole continue. Ainsi, un TER moyen et une distribution du NC peut être affectée à chaque problème de reconnaissance.

Lorsque l'utilisateur virtuel communique avec le modèle de système ASR, il lui communique un vecteur de concepts en accord avec le but qu'il s'est fixé. A la réception de ce vecteur, le modèle ASR réalise l'algorithme suivant :

```

- Pour chaque concept de la liste
{
  > Choisir aléatoirement un réel entre 0 et 1
  > Si R < TER(concept courant) // erreur ASR
  {
    • Substituer le concept courant avec un
      autre de même nature (simule l'erreur de
      reconnaissance)
    • Emettre un NC partiel grâce à la
      courbe des mots mal reconnus de la
      distribution appropriée au concept.
  }
  > Sinon // pas d'erreur
  {
    • Transmettre concept courant sans
      modification
    • Emettre un NC partiel grâce à la
      courbe des mots bien reconnus de la
      distribution appropriée au concept.
  }
}
- Transmettre le nouveau vecteur à l'agent
- Emettre un NC global pour le vecteur en
  multipliant tous les NC partiels.

```

Il est possible de simuler la production des N meilleurs vecteurs reconnus étant donné le vecteur initial en itérant cet algorithme. Ceci permettrait, entre autres, dans d'autres applications, de générer des dialogues plus réalistes comportant des demandes de confirmation plus ergonomiques dans lesquelles le système ne demanderait pas simplement de répéter la dernière phrase mais

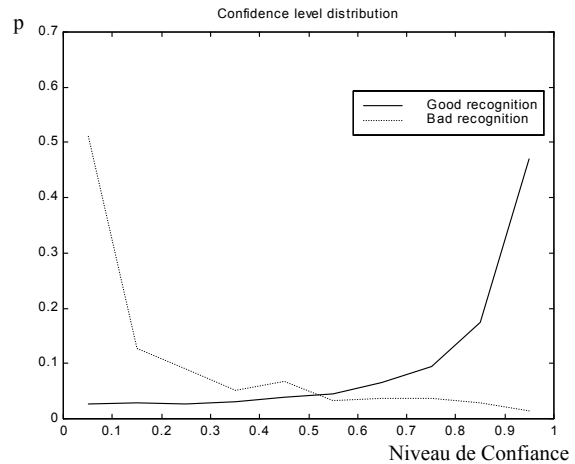


Figure 3: Exemple de distribution du NC proposerait plusieurs alternatives de réponse en fonctions des résultats de reconnaissance ayant obtenu les meilleurs scores.

3.4. Signal de Renforcement

Le signal de renforcement est construit de manière à exprimer le coût instantané d'une action prise dans un état donné. Il sera calculé, dans notre cas, comme la somme pondérée de différents termes :

$$r_t = W_t N_t + W_{dba} N_{dba} + W_{pr} N_{pr} - W_{NC} NC - W_s f(U_s)$$

où

- N_t est égal à 0 si $s_{t+1} = s_F$ et à 1 dans les autres cas.
- N_{dba} est le nombre de requêtes à la base de données
- N_{pr} est le nombre de données présentées à l'utilisateur
- NC est le niveau de confiance du vecteur courant
- $f(U_s)$ est une fonction de la satisfaction de l'utilisateur
- W sont des poids positifs ajustables

Ainsi, la minimisation du coût d'un dialogue obtenu en sommant les r_t donnera lieu à une stratégie établissant un compromis entre la longueur du dialogue, le nombre d'accès à la base de données, la quantité d'information délivrée à l'utilisateur, le niveau de confiance sur tout le dialogue et des données plus ou moins subjectives tels que la réalisation ou non du but, le mécontentement dû à des questions posées plusieurs fois par le système, etc. On décrira ces dernières sous le terme générique de "satisfaction" bien que les autres points (comme N_p ou N_t) puissent aussi être considérés comme faisant partie de la satisfaction générale de l'utilisateur.

4. EXPÉRIENCE

Cette méthode d'apprentissage a été appliquée sur le problème pratique de la vente d'ordinateur par téléphone. Plus de 350 types d'ordinateurs étaient consignés dans la base de données qui contenait 2 tables (pour les portables et les ordinateurs de bureau) de 6 champs chacune : pc_mac, processor_type, processor_speed, ram_size, hdd_size et brand. Le MDP correspondant à cette tâche peut être décrit comme suit :

Actions : On peut définir 5 types d'actions.

- GREETING : phrase d'accueil
- ASK(arg) : question contraignant l'argument **arg** (ex : "quel processeur voulez-vous ?" = ASK(processor_type)
- RELAX(arg) : question relaxant l'argument **arg** (ex : "La marque est-elle importante ?" = RELAX(brand))
- DBQUERY : requête à la base de données
- CLOSE : présenter les informations demandées et fermer le dialogue.

La valeur de **arg** peut être soit la table (laptop, desktop) ou un des 6 champs. Il y a donc 17 actions possibles.

Etats : Chaque état est représenté par 2 caractéristiques.

- Un vecteur de 7 booléens $[f_x]$. Un f_x est "vrai" si l'utilisateur a donné la valeur du $x^{\text{ème}}$ **arg**. (ex : si l'utilisateur a précisé s'il voulait un portable, f_0 est "vrai")
- Une information sur le niveau de confiance de chaque f_x . Ici, nous avons défini un seuil qui sépare les valeurs élevées du NC et les autres. Ainsi, il y a 2 valeurs possibles pour le NC : "HAUT" et "BAS".

Pour chaque valeur de **arg**, il existe donc 3 possibilités : $\{f_x = \text{faux}, \text{NC} = \text{undef}\}$, $\{f_x = \text{vrai}, \text{NC} = \text{BAS}\}$, $\{f_x = \text{vrai}, \text{NC} = \text{HAUT}\}$. Ceci nous mène à 3^7 états.

Algorithme de RL : Etant donné que l'agent ne doit interagir qu'avec un utilisateur virtuel durant la phase d'apprentissage, la méthode "Exploring Starts Monte Carlo" [5] a été choisie. En effet, l'agent n'a pas besoin de suivre une stratégie ayant un réel sens et peut se permettre d'explorer le plus possible l'espace des états.

5. RÉSULTATS

La figure 4 décrit la stratégie finale, qui semble être la stratégie optimale, apprise par le système. Cette stratégie est obtenue après plusieurs milliers de dialogues simulés et est la dernière d'une série de stratégies sub-optimales par lesquelles le système est passé. Elle peut être expliquée comme suit : après avoir prononcé la phrase d'accueil, le système peut

- avoir obtenu suffisamment d'information de l'utilisateur (qui a précisé une description assez complète de son choix immédiatement). Dans ce cas, il peut effectuer une requête à la base de données.
- demander un complément d'information afin d'être plus sûr d'obtenir un résultat de requête intéressant.

Dans le cas où la requête ne donne aucun résultat, le système demande de relaxer certaines contraintes. Par contre, si la requête donne trop de résultats (et donc un nombre élevé de données présentables à l'utilisateur), le système essaie de contraindre un peu plus celle-ci.

Lorsque l'agent doit poser des questions contraignantes, il procède dans un ordre particulier. En effet, vu l'introduction du niveau de confiance dans le signal de renforcement, le système préfère poser des questions qui lui vaudront a priori des réponses à NC plus élevés et qui résulteront en un niveau de confiance global plus grand pour le dialogue entier. De même, cela impliquera une diminution de la longueur du dialogue. Ainsi, l'agent posera préférentiellement une question portant sur des

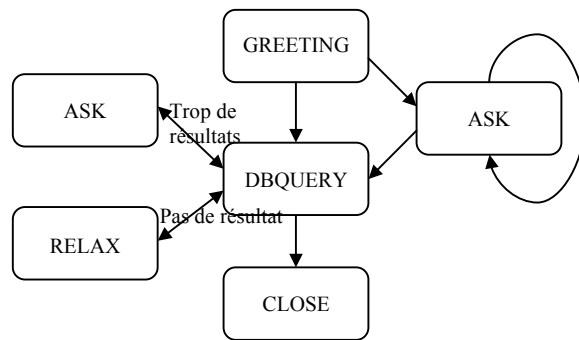


Figure 4 : Stratégie Finale. Chaque état, représenté par un rectangle, est associé à son action optimale

nombre, par exemple, qui donneront probablement lieu à moins d'erreur de reconnaissance. Il préférera donc une question concernant la taille de la mémoire désirée plutôt que la marque d'ordinateur.

6. CONCLUSION ET PERSPECTIVES

L'intérêt majeur de l'insertion d'un modèle de système de reconnaissance dans l'environnement d'apprentissage est évidemment que le système évite, si possible, de poser des questions dont la réponse va éventuellement donner lieu à des sous-dialogues de confirmation, à des requêtes de bases de données erronées et rallonger inutilement le dialogue. De plus, cela évite de devoir poser plusieurs fois une question similaire à l'utilisateur, ce qui augmente sa satisfaction. Dans le futur, il serait intéressant de simuler et d'introduire des niveaux de confiance de plus haut niveau qui tiendraient compte, par exemple, du contexte en tenant compte d'un niveau de confiance traduisant la cohérence d'une phrase reconnue avec l'état du dialogue dans lequel le système se trouve.

BIBLIOGRAPHIE

- [1] R. Pieraccini, E. Levin, W. Eckert (2000) "A Stochastic Model of Human Machine Interaction for Learning Dialog Strategies". *IEEE Transactions on Speech and Audio Processing*, Vol. 8 pp 11-23
- [2] R. Pieraccini, E. Levin, W. Eckert (1997) "User Modeling for Spoken Dialogue Systems" *Proc. IEEE ASR Workshop, Santa Barbara*
- [3] S. Young, K. Sheffler, (2000) "Probabilistic Simulation of Human-Machine Dialogues" *Proc. ICASSP, Istanbul, Turkey*
- [4] E. Mengusoglu, C. Ris, (1998) "Use of Acoustic Prior Information for Confidence Measure in ASR Applications", *Eurospeech 2001 Scandinavia, Aalborg*
- [5] R.S. Sutton, A.G. Barto (1998) "Reinforcement Learning : An Introduction" *MIT Press*