Modèles de langage hiérarchiques pour les applications de dialogue en parole spontanée

F. Béchet, Y. Estève, R. De Mori
LIA - Université d'Avignon - BP1228 - Avignon Cedex 9

1 Résumé - Abstract

Le cadre de cette étude [1] concerne les systèmes de dialogue via le téléphone entre un serveur de données et un utilisateur. Nous nous intéresserons au cas de dialogues non contraints où l'utilisateur à toute liberté pour formuler ses requêtes. Généralement, le module de Reconnaissance Automatique de la Parole (RAP) de tels serveurs utilise un seul Modèle de Langage (ML) de type bigramme ou trigramme pour modéliser l'ensemble des interventions possibles de l'utilisateur. Ces ML sont appris sur des corpus de phrases retranscrites à partir de sessions entre le serveur et plusieurs utilisateurs. Nous proposons dans cette étude une méthode de segmentation de corpus d'apprentissage de dialogue utilisant une stratégie mixte basée à la fois sur des connaissances explicites mais aussi sur l'optimisation d'un critère statistique. Nous montrons qu'un gain en terme de perplexité et de taux d'erreurs/mot peut être constaté en utilisant un ensemble de sous modèles de langage issus de la segmentation plutôt qu'un modèle unique appris sur l'ensemble du corpus.

Within the framework of Human-Computer dialogue in spontaneous speech, we propose in this paper a method which automatically builds, from a training corpus, a set of Language Models (LMs) organized as a binary tree. Each LM correspond to a specific dialogue state, where the general LM is attached to the root node and the more specialized ones are represented by the leaves. Such LMs can be used to automatically adapt the decoding process to the dialog situation performed. We propose a two-pass decoding strategy, which implements this idea by dynamically selecting a set of LMs according to the dialog situation detected.

Reconnaissance Automatique de la Parole ; Modèles de Langage statistique ; Serveurs de Dialogue ; Arbre de Décision

2 Introduction

Les Modèles de Langage (ML) utilisés habituellement dans les systèmes de reconnaissance automatique de la parole sont basés sur une approche probabiliste nécessitant un apprentissage sur corpus. Cet apprentissage permet d'estimer les probabilités de transition entre les mots d'une même phrase. Durant la reconnaissance, l'historique utilisé pour prédire le prochain mot est généralement réduit aux deux ou trois mots précédents (modèles bigrammes ou trigrammes).

[1] Ces travaux sont réalisés en collaboration avec France-Telecom R&D sous le contrat 971b427

Si ces caractéristiques sont communes à l'ensemble des systèmes, le nombre de ML utilisé et leurs combinaisons dépendent fortement de l'application visée. Dans le cas d'un système de dictée généraliste, il est bien difficile de segmenter a priori les différents cas d'utilisation du système : un seul ML, mélangeant thèmes et types de phrases, est généralement employé.

Dans les applications de dialogue pour les serveurs téléphoniques, il est en revanche possible de segmenter le corpus d'apprentissage du ML selon différents états de dialogue correspondant aux interactions entre le serveur et l'utilisateur. Cependant il reste à déterminer d'une part les situations de dialogue les plus discriminantes du point de vue de la reconnaissance et d'autre part la méthode de sélection et de combinaison des ML obtenus sur chacun des sous-corpus d'apprentissage.

Ces travaux proposent des réponses à ces questions en présentant une méthode de classement d'un corpus de dialogue en sous-corpus basée sur une approche mixte : des connaissances linguistiques explicites sont tout d'abord utilisées pour segmenter le corpus d'apprentissage du ML. Puis, une technique de classification statistique basée sur des arbres de décision permet de segmenter à nouveau chacun des sous-corpus selon un critère largement utilisé en RAP : la perplexité. Enfin, nous proposons un ML hiérarchique codant l'ensemble des sous-ML à l'intérieur d'une structure d'arbre utilisable dans un décodage à deux passes.

3 Segmentation de corpus

Les corpus d'apprentissage des ML utilisés dans les systèmes de RAP pour les applications de dialogue téléphonique sont constitués de retranscriptions orthographiques de sessions de dialogue entre un utilisateur (naïf ou expert) et le serveur de dialogue. Ces sessions correspondent au traitement d'une requête complète d'un utilisateur, aboutissant ou non à un succès, composées de questions et de réponses entre le serveur et l'utilisateur. Si le serveur admet un dialogue ouvert, il est laissé toute liberté à l'utilisateur pour formuler ses requêtes. Dans ce cas, le système doit être à même de pouvoir traiter les phénomènes inhérents à la parole spontanée, tels que les phrases agrammaticales, les hésitations, les reprises, etc. Cette extrême variabilité est compensée par, d'une part un domaine sémantique généralement restreint (horaire de train, programme de cinéma, etc.), d'autre part des régularités fortes dans les structures de phrases liées à certaines situations de dialogue. Ainsi, les premières requêtes exprimées par un utilisateur commencent généralement par des patrons de phrases tels que : *je + [voudrais , recherche , désire , . . .]*

Les diverses situations de dialogue qui peuvent être rencontrées dans une session entre un utilisateur et un serveur dépendent bien évidemment de l'application visée. Cependant, à un niveau d'abstraction élevé, il est possible de distinguer des *macro-classes* de situations de dialogue indépendantes du serveur et du domaine d'application.

Avant de présenter notre méthode de découpage de corpus utilisant ces macro-classes, nous présentons les données sur lesquelles ont été effectuées toutes nos expériences :

Nous avons utilisé, dans cette étude, le corpus AGS (Sadek *et al.*, 1996) constitué de transcriptions de dialogue entre plusieurs utilisateurs et un serveur téléphonique. Deux domaines d'application sont visés par le serveur AGS : les informations météorologiques et la recherche de petites annonces d'emploi. Le corpus d'apprentissage est constitué de 9842 phrases prononcées par plusieurs locuteurs et couvrant les deux domaines d'application. Le corpus de test est com-

posé de 1419 phrases représentant plusieurs sessions et plusieurs locuteurs différents. Le vocabulaire utilisé dans le corpus d'apprentissage contient 823 mots.

3.1 Segmentation en macro-classes

Une étude détaillée du corpus d'apprentissage permet de regrouper les interventions de l'utilisateur en différentes classes. Nous avons choisi, dans un premier temps, d'isoler des macro-classes indépendantes de l'état d'avancement du dialogue et des interventions de la machine. Ce choix s'explique essentiellement par le faible nombre de sessions complètes disponibles dans le corpus d'apprentissage. Ces macro-classes sont déterminées à partir de la structure des phrases prononcées par l'utilisateur, structure représentée sous la forme de patrons syntaxiques et lexicaux.

Les phrases du corpus d'apprentissage sont tout d'abord étiquetées et lemmatisées à l'aide d'un tagger morpho-syntaxique statistique (Spriet & El-bèze, 1995). Une analyse syntaxique partielle est ensuite effectuée sur le corpus étiqueté afin d'extraire les syntagmes composant les phrases. Par exemple, la phrase : *"J'aurais aimé connaître le numéro de téléphone ..."* sera étiquetée :

[(J',PPER) (aurais,AA) (aimé,VRPAS) (connaître,VINF)]
[(le,DET) (numéro,N) (de,PREP) (téléphone,N)]

Nous avons déterminé quatre macro-classes à partir du corpus d'apprentissage : $REQUETE$, contient l'ensemble des phrases représentant une première demande de l'utilisateur juste après le prompt du système ; $QUESTION$, contient les questions posées par l'utilisateur après une réponse du serveur ; $REPONSE$, contient les réponses de l'utilisateur à une question du serveur ; $AUTRE$, contient ce qui n'a pu être étiqueté dans les autres classes. On trouve ici des interventions de gestion du dialogue (ex : *"annulation"*, *"au revoir"*, *"merci"*) des phénomènes extra linguistiques (ex : *"ah"*, *"hum"*, *"euh"*) des phrases tronquées ou encore des mouvements d'humeur suite à une mauvaise interprétation du dialogue par la machine (ex : *"laisse tomber"*, *"c'est pas grave"*).

Un ensemble de règles basées sur la présence de certains mots, de certaines classes syntaxiques ou encore de certaines structures syntaxiques ont été écrites afin de classer les phrases d'apprentissage dans chacune de ces catégories.

3.2 Segmentation hiérarchique

Cette méthode consiste à raffiner la première segmentation du corpus en macro-classes en optimisant, de manière itérative, un paramètre influençant directement la reconnaissance : la perplexité. C'est une mesure communément employée pour estimer les performances d'un modèle de langage indépendamment de l'aspect acoustique de la reconnaissance. Bien qu'il n'existe pas de lien formel liant les performances de reconnaissance et la mesure de perplexité, ces deux quantités suivent généralement des évolutions communes. En mesurant la perplexité d'un modèle sur un texte, on est à même d'apprécier la capacité du modèle à prédire ce même texte. Plus cette mesure est petite, plus le texte est proche du corpus utilisé pour apprendre le modèle.

Durant cette phase de segmentation nous allons utiliser une structure d'arbre binaire où deux informations sont associées à chaque noeud : une expression régulière et un sous-corpus sat-

isfaisant l'expression régulière du noeud. Un noeud peut générer deux fils si une extension de l'expression régulière courante permet de scinder le sous-corpus en deux et si des modèles de langage appris sur ces deux sous-corpus fils obtiennent des perplexités plus faibles que celle obtenue avec le modèle de langage du noeud père. Cette méthode est inspirée des Arbres de Classification Sémantiques introduit par (Kuhn & de Mori, 1996).

Contrairement aux problèmes de classification résolus habituellement par des méthodes à base d'arbres de décision, nous ne connaissons pas dans notre cas le nombre optimal de classes à discerner. Le nombre minimal correspond au nombre de macro-classes déjà présentées. Le nombre maximal de classes est borné à la fois par la taille de corpus nécessaire à l'apprentissage d'un modèle bigramme et par le gain minimum en perplexité requis entre le noeud père et les deux noeuds fils.

Le seuil choisi sur la taille minimale d'un sous-corpus permet de contrôler la taille de l'arbre. La figure 1 présente les premières branches d'un exemple d'arbre obtenu sur le corpus d'apprentissage AGS en fixant une taille minimale de 300 phrases pour chaque noeud. Cet arbre contient 43 noeuds internes et 45 feuilles, ce qui représente un ensemble de 88 sous-corpus et sous-ML.

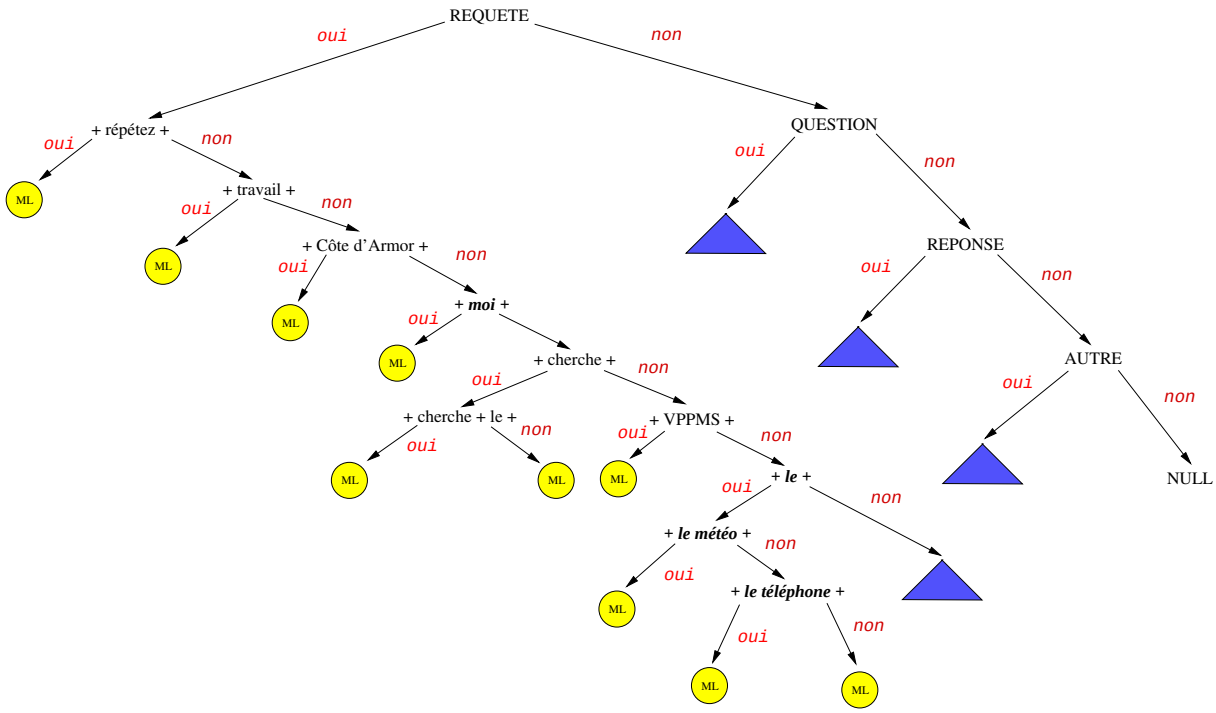

Figure 1: Exemple d'arbre de sous-ML

3.3 Evaluation de la segmentation hiérarchique

Nous avons évalué l'arbre de sous-ML à travers deux aspects : les gains en perplexité obtenus sur le corpus de test et l'évolution du taux d'erreurs/mot en utilisant le modèle général et les sous-ML (tableau 1). Pour cela, nous avons analysé les phrases retranscrites du corpus de test avec les expressions régulières liées aux noeuds de l'arbre et nous avons déterminé à

quelle feuille chacune des phrases appartenait. Enfin, nous les avons décodées avec le sous-ML $ML_{feuille}$ correspondant à la feuille choisie.

Même si les gains obtenus en reconnaissance sont décevants par rapport aux gains constatés en perplexité, ils sont néanmoins significatifs et valident l'apport d'une segmentation préalable du corpus d'apprentissage en différents sous-corpus. Il n'est cependant pas évident de choisir a priori, lors de la reconnaissance, l'un des sous-modèles uniquement à partir de l'historique du dialogue en cours. De plus, il se peut que certaines situations du test correspondent à plusieurs sous-corpus de l'apprentissage. Pour ces raisons, nous présentons maintenant une méthode de sélection dynamique de sous-ML basée sur une exploration hiérarchique de l'arbre contenant les ML.

classe	PP$_{général}$	PP$_{feuille}$	gain	Err$_{général}$	Err$_{feuille}$	gain
$REQUETE$	$9,83$	$7,68$	$21,89\%$	$19,39$	$17,38$	$10,36\%$
$REPONSE$	$16,82$	$10,71$	$36,34\ \%$	$24,7$	23.79	$3,7\%$
$QUESTION$	$13,55$	$8,72$	$35,62\ \%$	$26,9$	$25,49$	$5,21\%$
$AUTRE$	$25,44$	$9,93$	$39,05\%$	$56,32$	$50,12$	11%

Table 1: Résultats en perplexité et taux d'erreurs/mot par classe de mots sur la segmentation hiérarchique

4 Sélection dynamique de sous-modèles de langage

Les systèmes de reconnaissance disposant de plusieurs ML utilisent généralement deux stratégies : soit les ML sont combinés à l'aide de coefficients fixes ou dynamiques (Kalai *et al.*, 1999) ; soit un processus de sélection intervient avant la reconnaissance pour déterminer dans quelle situation le système se trouve (d'un point de vue thématique ou du point de vue de l'historique du dialogue) et le ML est choisi en conséquence (Riccardi & Gorin, 2000).

Notre méthode de sélection dynamique emprunte à chacune de ces méthodes : d'un coté notre arbre de sous-ML contient déjà des combinaisons de ML, et il suffit de parcourir une branche depuis la racine jusqu'à une feuille pour spécialiser de plus en plus la reconnaissance ; d'un autre coté nous allons choisir un sous-modèle en parcourant l'arbre grâce à un processus de sélection utilisant les résultats d'un premier décodage. Cette méthode, inspirée des méthodes d'adaptation de modèles acoustiques aux caractéristiques d'un locuteur, permet d'adapter le ML à la situation de dialogue détectée.

Dans cette méthode, un premier décodage est effectué en utilisant le ML général associé à la racine de l'arbre. Cette première hypothèse, appelée H_1, va nous servir à sélectionner un noeud de l'arbre dont le sous-ML correspondant sera utilisé pour effectuer un deuxième décodage et ainsi produire l'hypothèse H_2. La méthode de sélection consiste simplement à parcourir l'arbre, depuis la racine, en choisissant à chaque noeud le fils qui offre le sous-ML ayant la perplexité la plus basse sur H_1. Lorsqu'on arrive à une feuille, ou lorsque la descente d'un noeud en son fils n'offre pas un gain de perplexité sur H_1, l'algorithme s'arrête et le sous-ML attaché au noeud courant est sélectionné pour effectuer le second décodage. En procédant ainsi, on ne fait aucune hypothèse sur l'état du dialogue, et la structure hiérarchique de l'arbre permet de mélanger divers ML pour traiter n'importe quelle phrase prononcée par l'utilisateur.

Pour évaluer la méthode de sélection dynamique, nous avons dans un premier temps calculé toutes les hypothèses H_1 correspondant aux phrases du corpus de test. Pour chacune d'entre elle, nous avons sélectionné le noeud N avec la méthode présentée au paragraphe précédent. Pour chaque noeud i sur le chemin de la racine à N, nous avons évalué les performances du sous-ML ML_{noeud_i} en terme de taux d'erreurs/mot sur l'ensemble du corpus de test. Plus i augmente, plus le sous-ML se spécialise, et l'optimum devrait se trouver en N. Les résultats sont présentés dans le tableau 2. Il est intéressant de constater que la courbe respecte le comportement attendu : le taux d'erreurs/mot diminue au fur et à mesure de la descente dans l'arbre. Même si on n'obtient pas les gains présentés dans le paragraphe 3.3 à cause des erreurs contenues dans H_1, les gains sont toutefois réguliers et semblent justifier la méthode. Cependant une marge d'amélioration importante réside dans la méthode de sélection pour s'approcher de la sélection optimale.

prof.	0	1	2	4	6	8	10	12
err.	24.7	24.5	24.2	24.1	23.9	23.8	23.8	23.7

Table 2: Evolution du taux d'erreurs/mot en fonction de la profondeur dans l'arbre

5 Conclusion

Les méthodes de segmentation de corpus présentées répondent aux questions posées dans l'introduction : en complétant une méthode basée sur l'étude de situations de dialogue avec une méthode utilisant un critère purement statistique, nous montrons qu'un gain significatif, à la fois en perplexité et en taux d'erreurs/mot peut être obtenu à condition de choisir le sous-modèle de langage adéquat. Notre méthode de sélection dynamique permet d'adapter la reconnaissance à la phrase prononcée en répondant au problème du choix du ML. Nous pensons cependant que les résultats obtenus pourraient être améliorés en augmentant la robustesse de la méthode de sélection aux erreurs de reconnaissance des hypothèses H_1.

Références

KALAI A., CHEN S., BLUM A. & ROSENFELD R. (1999). On-line algorithms for combining language models. In *ICASSP*.

KUHN R. & DE MORI R. (1996). The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(5), 449–460.

RICCARDI G. & GORIN A. L. (2000). Stochastic language adaptation over time and state in natural spoken dialogue systems. *IEEE Transactions on Speech and Audio*, **8,1**.

SADEK D., FERRIEUX A., COZANNET A., BRETIER P., PANAGET F. & SIMONIN J. (1996). Effective human-computer cooperative spoken dialogue: the ags demonstrator. In *ICSLP'96, USA*.

SPRIET T. & EL-BÈZE M. (1995). Etiquetage probabiliste et contraintes syntaxiques. *Traitement Automatique des Langues*, **Vol 36, n1-2**.