

INTRODUCTION

1 Contexte de l'étude

Un système de reconnaissance est un système capable de traduire ou de décoder un signal acoustique de parole émis par un locuteur, et reçu à travers un capteur, en un mot ou une séquence de mots. L'utilité d'un système de reconnaissance est montrée à travers les tâches qu'on veut réaliser sans pour cela utiliser nos mains ou nous déplacer. On peut citer parmi ces tâches : la réservation de billets de train ou d'avion, la commande vocale pour handicapés, etc.

La plupart des systèmes de reconnaissance utilisés aujourd'hui, emploient pour la phase de décodage une méthode statistique. Cette dernière est à base de modèles de Markov cachés. Ces modèles ont eu beaucoup de succès grâce aux taux de reconnaissance élevés obtenus aujourd'hui. Cependant ce taux de reconnaissance chute lorsque le système est mis en exploitation (locuteur peu coopératif, bruit ambiant, etc.). Que faut-il faire dans ce cas ?

Faut-il imposer à l'utilisateur d'un tel système un cahier de recommandations à suivre du style : éloignez votre chien, ne touchez pas, ne criez pas, ne vous énervez pas, etc. A coup sûr le système de reconnaissance réalisé sera mis aux oubliettes.

Une autre approche est d'améliorer les performances du système de reconnaissance en évaluant l'apport de la modélisation de l'application (les contraintes syntaxiques), de la détection bruit/parole, de la reconnaissance dans un environnement bruité, de la recherche des N meilleures solutions, etc.

Cette thèse a débuté en Mars 1991 et avait pour objectif le développement d'une méthode de recherche des N meilleures solutions pour le système de reconnaissance du CNET et la mise en oeuvre des post-traitements.

La recherche des N meilleures solutions implique la mise en oeuvre de techniques permettant au système de reconnaissance de développer non plus une solution unique (la meilleure au sens des probabilités) mais les N meilleures solutions possibles pour un mot prononcé (ou la phrase). Ensuite un post-traitement se charge de retrouver le mot prononcé (ou la phrase) dans la liste proposée.

2 Description de l'étude

Une application donnée est décrite par une syntaxe. [Soong, 91] proposa de rechercher les N meilleures solutions dans ce réseau syntaxique. Sa méthode consiste à utiliser l'algorithme de Viterbi dans la phase aller (phase synchrone) et l'algorithme A* dans la phase retour (phase asynchrone). L'algorithme de Viterbi se charge de calculer et de mémoriser les différentes probabilités d'émission des mots sur le réseau syntaxique à chaque instant. L'algorithme A* est utilisé pour remonter le chemin à partir du dernier noeud syntaxique. L'estimation

correspondant à la portion du signal non encore traitée pendant la phase retour provient des valeurs mémorisées pendant la phase aller. La solution déclarée correcte est retrouvée à partir d'un post-traitement syntaxique.

Cependant dans le cas d'une application de reconnaissance de mots isolés ou bien de mots connectés (les nombres par exemple), la syntaxe de l'application est très rudimentaire. De ce fait un post-traitement syntaxique ne peut avoir lieu. Par contre on peut envisager un autre type de post-traitement reposant sur une approche segmentale. Pour cela, il faut que le système de reconnaissance fournisse en plus des N meilleures solutions, leurs alignements respectifs (segmentations "phonétiques"). Pour obtenir ces alignements, nous proposons d'adapter la méthode de [Soong, 91] pour la recherche des N meilleures solutions au niveau acoustique (au lieu du niveau syntaxique). Cela se traduit par une complexité supplémentaire dans le développement des N meilleures solutions en un temps raisonnable.

Les N meilleures solutions obtenues pour chaque mot (ou phrase) prononcé sont des solutions syntaxiquement différentes et non pas des alignements différents d'une même solution. Ainsi chaque solution fournie va correspondre à une observation. Et toutes ces observations diffèrent par leur segmentation. Pour un mot prononcé, on ne doit donc plus traiter une observation unique mais N observations issues des N meilleures solutions.

Pour retrouver la solution correcte, on va appliquer un post-traitement segmental. Les post-traitement segmentaux développés [Phillips, 91], [Austin, 92] jusqu'à aujourd'hui font intervenir les réseaux connexionnistes. Dans notre approche, on va montrer qu'il est possible de rester dans le cadre d'une modélisation statistique, ainsi on va développer une nouvelle approche utilisant un post-traitement segmental statistique.

On a mentionné que chaque solution proposée constitue une observation qui diffère des autres observations par sa segmentation. Les segments d'une solution sont déclarés corrects, si la solution obtenue correspond au mot prononcé. Dans le cas contraire ces segments sont déclarés incorrects. Ainsi sont définis deux types d'étiquettes : correcte et incorrecte. Pour chacune d'elle est associé un modèle statistique. Le modèle des corrects correspond à une segmentation correcte et le modèle des incorrects correspond à une segmentation incorrecte.

Deux modèles statistiques sont donc associés à chaque segment : le premier représente les statistiques des segments étiquetés corrects et le second modèle représente les statistiques des segments étiquetés incorrects. Le score d'un segment est obtenu en tenant en compte des deux modèles statistiques (corrects et incorrects). Le score segmental d'une solution est une combinaison des scores des segments la constituant et le score final est une combinaison du score segmental et du score markovien. La réponse fournie par le système est alors celle ayant le score final le plus élevé.

Dans le cadre d'une application de réservation de billet d'avion [Charpentier, 92] ou de train [Gitton, 92], par exemple, pour réduire le taux d'erreur il peut être intéressant de faire épeler les noms de villes par le locuteur. Dans ce cas, on dispose dans un dictionnaire des différentes villes possibles. Cette information va servir dans le cadre d'un post-traitement syntaxique. En effet, la première des N solutions développées qui correspond à une ville du dictionnaire est la ville reconnue.

3 Plan

Cette thèse est divisée en trois parties. La première partie présente la reconnaissance de la parole et les algorithmes de recherche dans un graphe. La seconde partie présente l'algorithme mis en oeuvre pour la recherche des N meilleures solutions et les résultats obtenus. Finalement la troisième partie présente un post-traitement syntaxique, puis le post-traitement segmental mis en oeuvre et analyse l'influence de divers paramètres dans la modélisation.

Première partie : Généralités

Tous les environnements liés à cette thèse sont décrits dans cette partie. Elle comporte 2 chapitres. Le premier chapitre donne un aperçu sur des systèmes de reconnaissance de la parole et décrit les différentes techniques utilisées pour la reconnaissance. On s'attardera plus sur la méthode statistique car c'est un des outils principaux de cette thèse. Enfin ce premier chapitre se termine par le système de reconnaissance développé au CNET qui sert de support à notre travail.

Vu que la méthode de recherche des N meilleures solutions est à base d'une méthode de recherche dans un graphe, on a trouvé intéressant de décrire dans le chapitre II les différentes méthodes de recherche dans un graphe, leurs avantages et leurs inconvénients. On décrit plus particulièrement l'algorithme A* utilisé dans la phase retour de la méthode de recherche des N meilleures solutions. Un exemple est traité pour comparer les différentes techniques de recherches dans un graphe.

Deuxième partie : Recherche des N meilleures solutions

Découpée en 4 chapitres, elle concerne la méthode de recherche des N meilleures solutions et les différents résultats obtenus. Le chapitre III donne une rétrospective générale sur les méthodes de recherche des N meilleures solutions développées jusqu'à aujourd'hui. Le chapitre IV présente l'adaptation de la méthode de [Soong, 91] pour la recherche des N meilleures solutions dans un réseau acoustique. On décrit notre façon de gérer la pile de recherche contenant les différents chemins pour avoir un temps de réponse le plus court possible.

Le chapitre V présente les différentes caractéristiques des différents corpus de données utilisées pour les recherches des N meilleures solutions ainsi que pour l'évaluation des post-traitements.

Le chapitre VI présente les différents résultats relatifs à cette deuxième partie. On décrit le développement d'un modèle de Markov, les contraintes imposées sur le nombre de solutions à développer, les résultats sur les différents corpus de données et finalement le temps de recherche des N meilleures solutions.

Troisième partie : Post-traitements

Elle concerne les différents post-traitements développés dans le cadre de cette thèse. On présente dans le chapitre VII, l'état de l'art dans les post-traitements. Différents types de post-traitements y sont décrits. On s'attarde sur le post-traitement segmental jumelant un modèle de Markov et une méthode connexionniste.

Le chapitre VIII présente le post-traitement syntaxique utilisé et les résultats obtenus en reconnaissance d'épellations. La recherche des N meilleures solutions utilise une syntaxe surgénérative tandis que le post-traitement utilise la vraie syntaxe de l'application pour déterminer la réponse du système.

Le chapitre IX présente notre approche segmentale statistique. On y décrit l'utilisation conjointe des modèles "corrects" et "incorrects" puis la détermination du score segmental de chaque solution. Finalement on présente et on analyse la combinaison linéaire des scores segmentaux et markoviens pour le choix de la réponse du système.

Finalement le chapitre X présente et analyse les résultats obtenus. On y montre l'apport de l'utilisation conjointe des modèles "corrects" et "incorrects", et on compare les performances du post-traitement statistique avec uniquement les modèles corrects à celles d'un post-traitement statistique avec des modèles corrects et incorrects.