

---



---

## CHAPITRE VII

### Etat de l'art dans les post-traitements

---

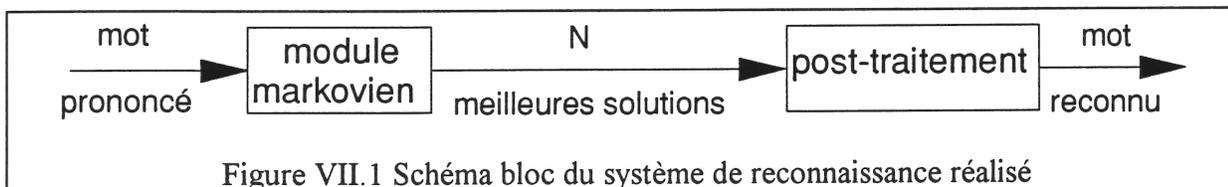


---

Ce chapitre commence par une vue générale sur les post-traitements existants aujourd'hui. Ensuite, on présente le post-traitement segmental statistique qui sera plus largement développé dans les chapitres IX et X.

### 1 Introduction

Le système de reconnaissance développé est composé de deux modules (figure VII.1). Le premier module propose les  $N$  meilleures solutions correspondant à une prononciation donnée d'un mot. Cependant la recherche des  $N$  meilleures solutions n'a d'intérêt que par la mise en oeuvre d'un post-traitement. Ce post-traitement est réalisé dans le second module. Son rôle est de déterminer la solution qui lui convient dans la liste proposée.



Quelles informations faut-il fournir à la sortie du module markovien, en plus de la liste des  $N$  meilleures solutions, pour permettre au module de post-traitement de retrouver la bonne solution ? La réponse à cette question est fonction de la nature du post-traitement. Différents types de post-traitements possibles sont décrits dans la suite de ce chapitre.

## 2 Caractéristiques des post-traitements

Les différents types de post-traitements généralement utilisés sont le post-traitement segmental [Phillips, 91], [Austin, 92], [Schwartz, 92], [Boiteau, 93-a], [Lokbani, 93-a], [Lokbani, 93-b], [Zavaliagos, 93] et le post-traitement syntaxique [Soong, 91], [Bates, 92], [Cardin, 92].

### 2.1 Post-traitement segmental

Le module markovien fournit pour chaque solution développée son alignement. L'alignement permet de déterminer les segments utilisés. Pour chaque segment, on extrait un ensemble de paramètres (durée d'un événement acoustique, spectre moyen sur un segment, ...). Le score segmental d'une solution est une combinaison des scores des segments la constituant. Le score final d'une solution tient compte du score segmental et du score markovien de cette solution. La solution ayant le score de combinaison le plus élevé est la solution retenue.

### 2.2 Post-traitement syntaxique

Beaucoup de syntaxes sont trop complexes pour être traitées directement. Une solution consiste à utiliser une syntaxe moins stricte et donc surgénérative. On recherche alors les  $N$  meilleures solutions qui correspondent à la phrase prononcée auxquelles on applique un post-traitement syntaxique afin de déterminer la solution correcte. Le post-traitement syntaxique consiste à utiliser une seconde syntaxe cette fois-ci plus contraignante (mais aussi plus exacte) et à rechercher parmi les  $N$  meilleures solutions envisagées la première qui respecte cette syntaxe. Les informations utilisées dans ce cas se résument à la liste des solutions proposées par le module markovien.

Cette seconde syntaxe peut être constituée par une liste fermée (exemples noms de villes [Jouvet, 93], numéros de cartes de crédits [Soong, 91], ...), où bien par une grammaire [Cardin, 92].

Une autre approche consiste à calculer des scores syntaxiques [Bates, 92] pour chaque solution. Ces scores proviennent de l'utilisation d'un modèle de langage. Les scores markoviens et syntaxiques sont combinés pour fournir le score final de chaque solution. De la même manière que dans l'approche segmentale, la solution ayant le score final le plus élevé est la solution retenue.

## 2.3 Post-traitements étudiés

Dans le cadre de la thèse, nous disposons de 3 types de corpus. Les corpus des mots isolés (chiffres et Trégor), le corpus des nombres à deux chiffres et le corpus des villes épelées.

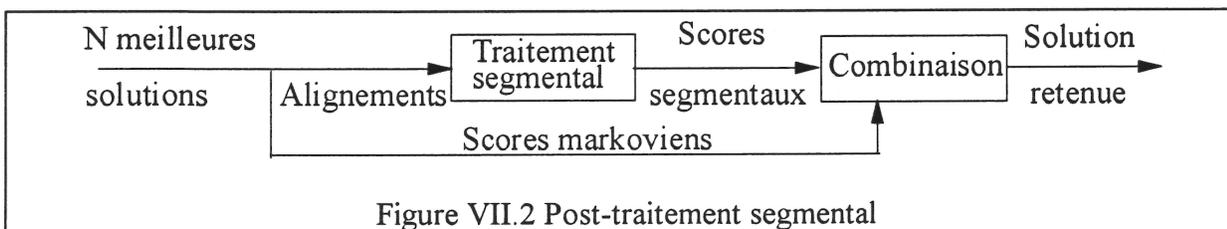
La reconnaissance des villes épelées nécessite l'introduction d'une syntaxe complexe dans la modélisation pour décrire toutes les villes possibles. Cette syntaxe, qui se résume à la liste des villes possibles, est difficile à intégrer à l'algorithme de décodage. Par contre son utilisation dans le cadre d'un post-traitement syntaxique est simple. Il suffit de vérifier si chaque solution appartient ou non à la liste.

Cependant dans le cas où il n'y a pas de contraintes syntaxiques complexes (chiffres, Trégor et nombres à deux chiffres), le post-traitement syntaxique n'est plus adapté. Dans ce cas il est utile d'utiliser le post-traitement segmental.

Nous avons donc développé les deux types de post-traitement : syntaxique et segmental.

## 3 Premiers pas dans le post-traitement segmental

Le principe du post-traitement segmental repose sur le calcul d'un nouveau score pour chaque solution proposée. Le choix de la bonne réponse s'effectue alors en tenant compte de ce nouveau score et du score obtenu par le module markovien. La figure VII.2 illustre les différents modules du post-traitement segmental.



Ce nouveau score (associé au post-traitement) doit être établi à partir d'informations les moins redondantes possibles avec celles traitées dans le module markovien, de manière à garder une certaine complémentarité dans les caractéristiques discriminantes extraites.

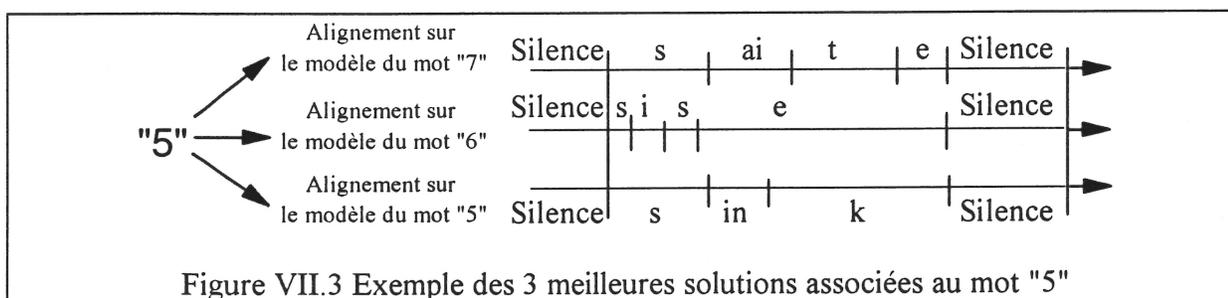
Le module markovien étant un système de reconnaissance synchrone à la trame d'analyse, la complémentarité souhaitée est assurée par le choix d'un traitement segmental de chaque solution.

Chaque solution est divisée en un nombre déterminé de segments. Un segment peut représenter un phonème, une syllabe ... ou une partie de ceux-ci. Sur chaque segment est calculé un score

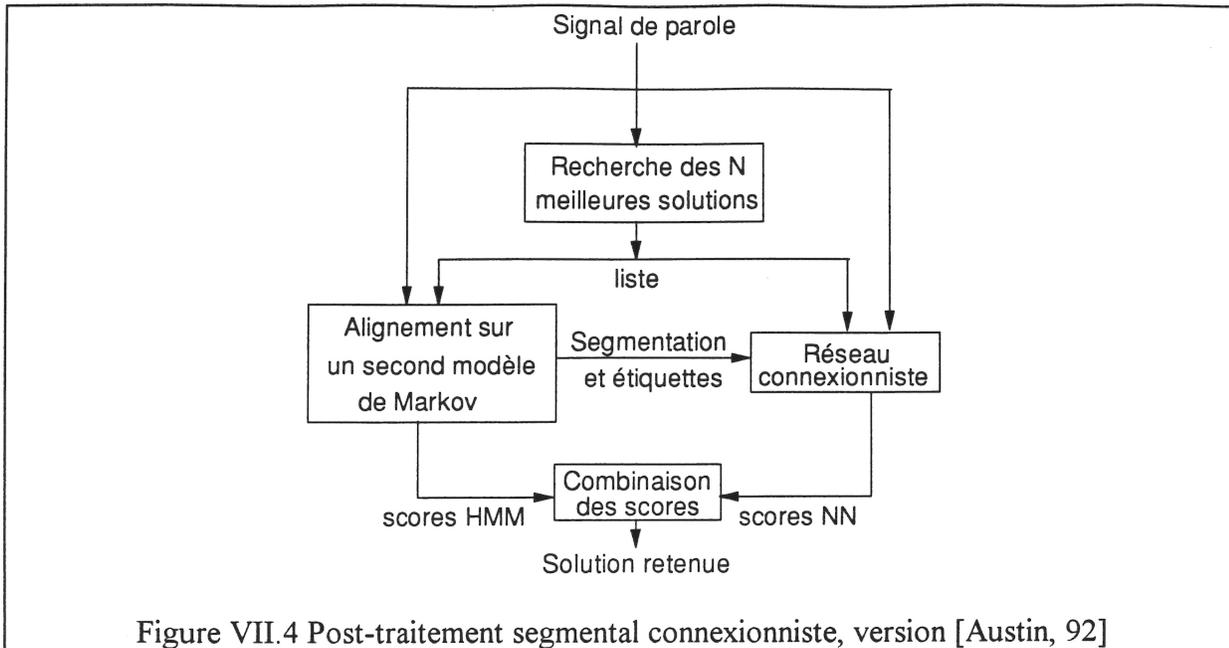
de post-traitement. Ce score prend en compte des informations non accessibles à un système de reconnaissance trame par trame telles que sa durée, son spectre moyen, son évolution spectrale, etc. Le score global d'un mot est alors obtenu par une recombinaison des scores de chacun des segments le constituant. Le calcul des scores de chacun des segments est réalisé par des méthodes statistiques [Lokbani, 93-a], [Lokbani, 93-b], ou par des méthodes connexionnistes [Phillips, 91], [Austin, 92], [Schwartz, 92], [Boiteau, 93-a], [Zavaliagos, 93].

### 3.1 Post-traitement segmental connexionniste

Le post-traitement segmental connexionniste a été introduit afin de valider ou d'invalider chacune des  $N$  solutions (hypothèses) proposées. Sur la figure VII.3 est présenté un exemple des 3 meilleures solutions (dans l'ordre "7", "6" et "5") obtenues pour le mot prononcé "5". Nous avons représenté en parallèle leur segmentation.



Nous remarquons que les seuls segments valides (i.e. qui correspondent aux segments du mot prononcé) sont "Silence de début", "s", "in", "k" et "Silence de fin". Les autres segments appartenant à des solutions non correctes sont déclarés non valides. De ce fait, il est intéressant d'utiliser une approche qui permette de distinguer entre ces étiquettes (i.e. valide et non valide). L'approche généralement employée dans ce cas est une approche connexionniste à cause de son pouvoir discriminant. Ainsi un réseau de neurones (ou plusieurs) est utilisé pour calculer d'une manière connexionniste le score d'un mot.



Dans le cas du travail présenté par [Austin, 92] et représenté sur la figure VII.4, un premier module à base d'un modèle de Markov peu détaillé (pas trop de paramètres) permet d'obtenir d'une manière rapide la liste des  $N$  meilleures solutions par la méthode de [Schwartz, 91]. Chaque solution de la liste est ensuite alignée sur un second modèle de Markov plus détaillé. Ce modèle fournit la segmentation [Ostendorf, 89], [Ostendorf, 91], l'étiquette et le score de chaque solution de la liste. La segmentation d'une solution est utilisée comme entrée d'un réseau connexionniste, la sortie correspond au score de chaque segment. Par la suite, le score global d'un mot est obtenu par multiplication des scores des segments le constituant.

Dans l'approche du CNET [Boiteau, 93-a], [Boiteau, 93-b], c'est le module de recherche des  $N$  meilleures solutions qui fournit avec chaque solution la segmentation associée. Deux réseaux connexionnistes sont utilisés, l'un pour calculer le score d'un segment et l'autre pour calculer le score d'un mot. Diverses architectures sont étudiées et comparées.

### 3.2 Post-traitement segmental statistique

Le post-traitement segmental statistique qui a fait l'objet de ce travail de thèse sera largement décrit dans les chapitres IX et X de ce document.

