# Reconnaissance Automatique de la Parole

## Présentation du module de Décodage acoustique

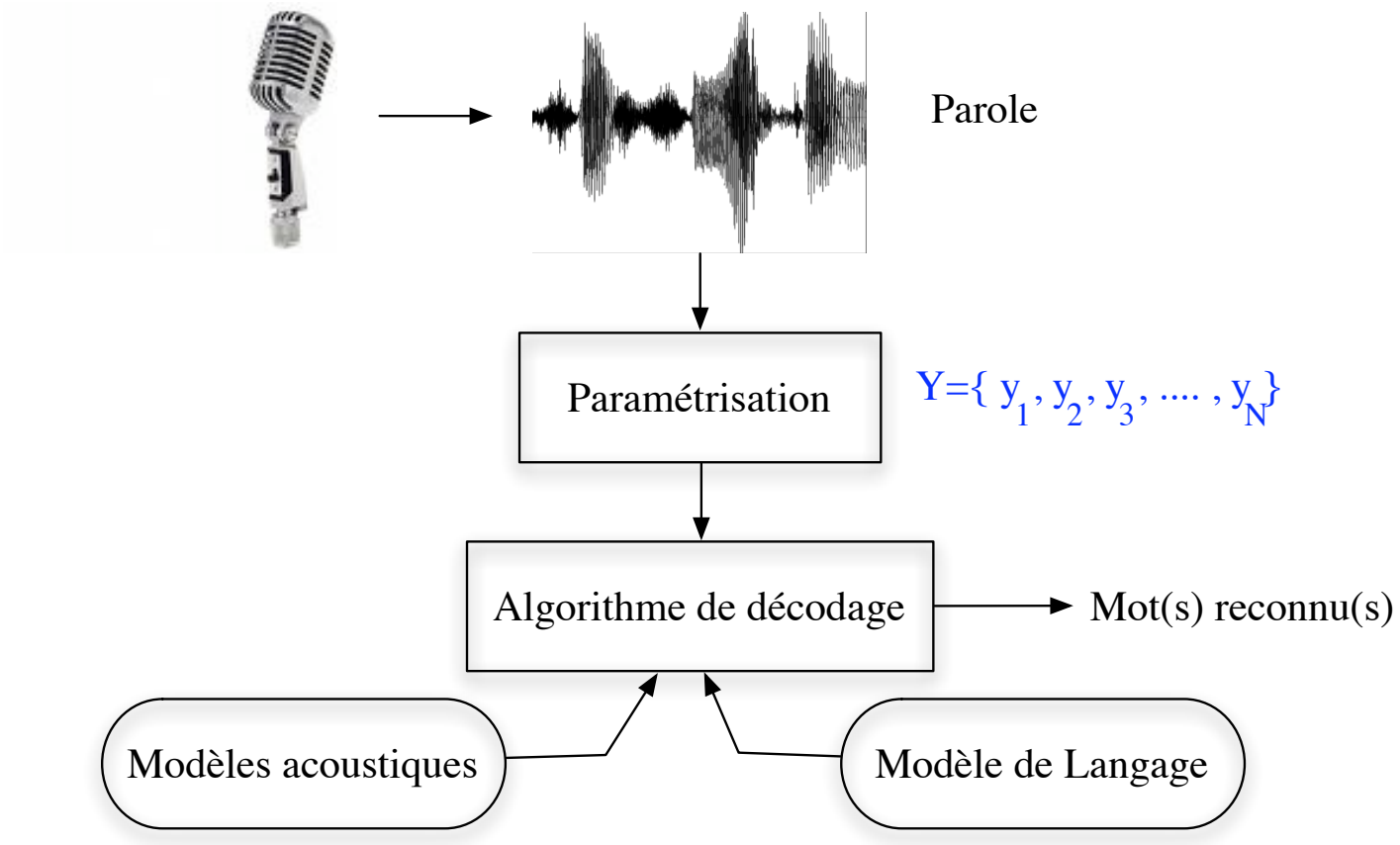
B. Jacob

IC2/LIUM

November 2, 2010

# Plan

# Vous êtes ici



# Segmentation du signal

On ne traite pas la totalité du signal

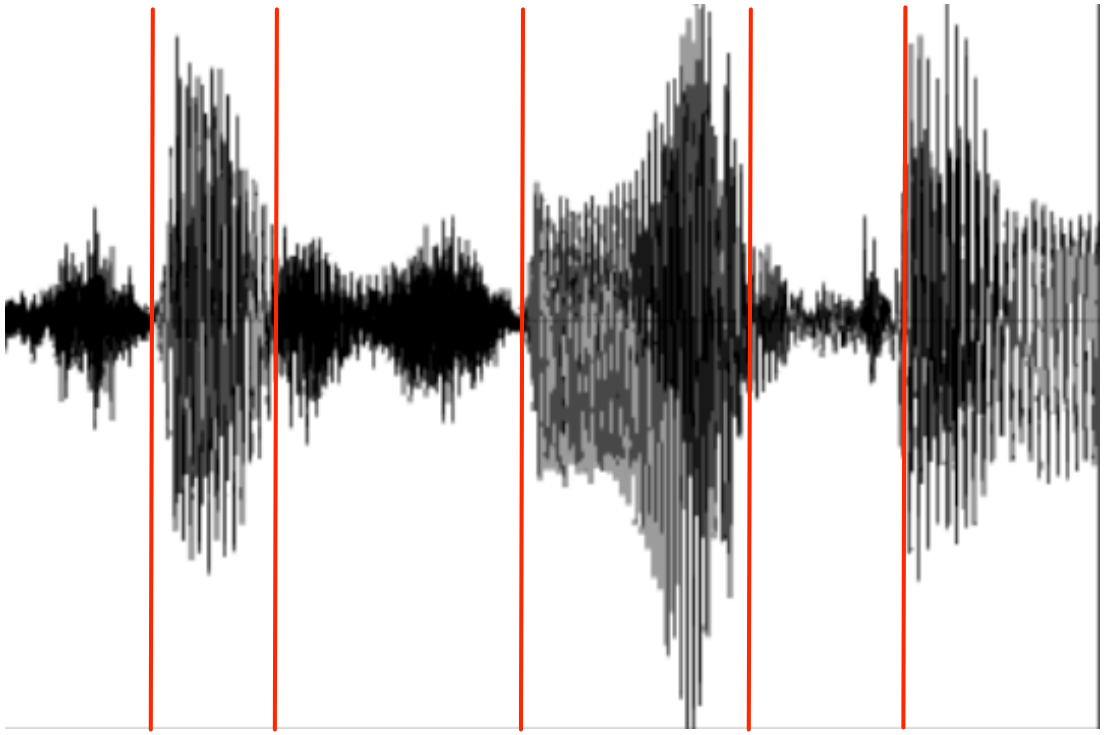
- information redondante
- volume de données trop important

→ "Découpage" du signal en segment

- de longueur variable (ex: recherche des parties stables et transitoires des sons [Obrecht, 88])
- de longueur fixe: échantillonnage à 16KHz (découpage centisecondes)

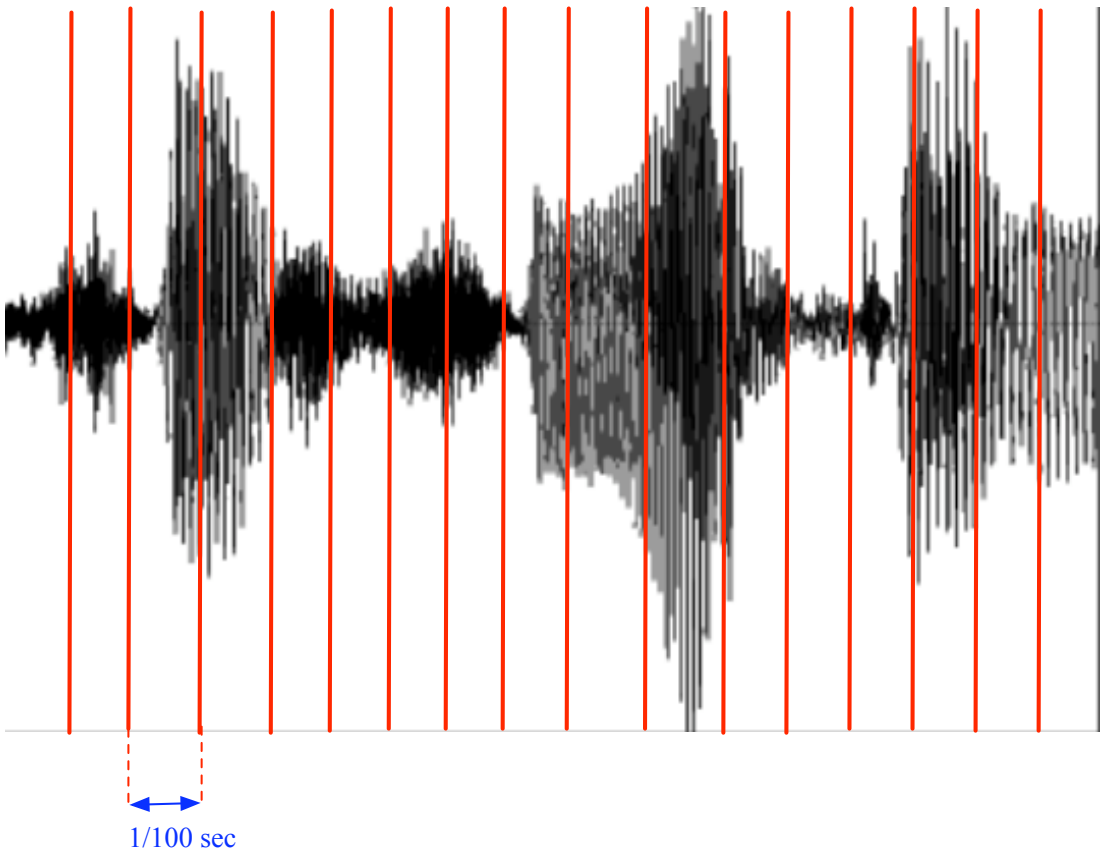
# Découpage segmental

- On recherche des segments de signal: voyelle, locuteur, silence, parole ...
- les segments sont de longueur variable



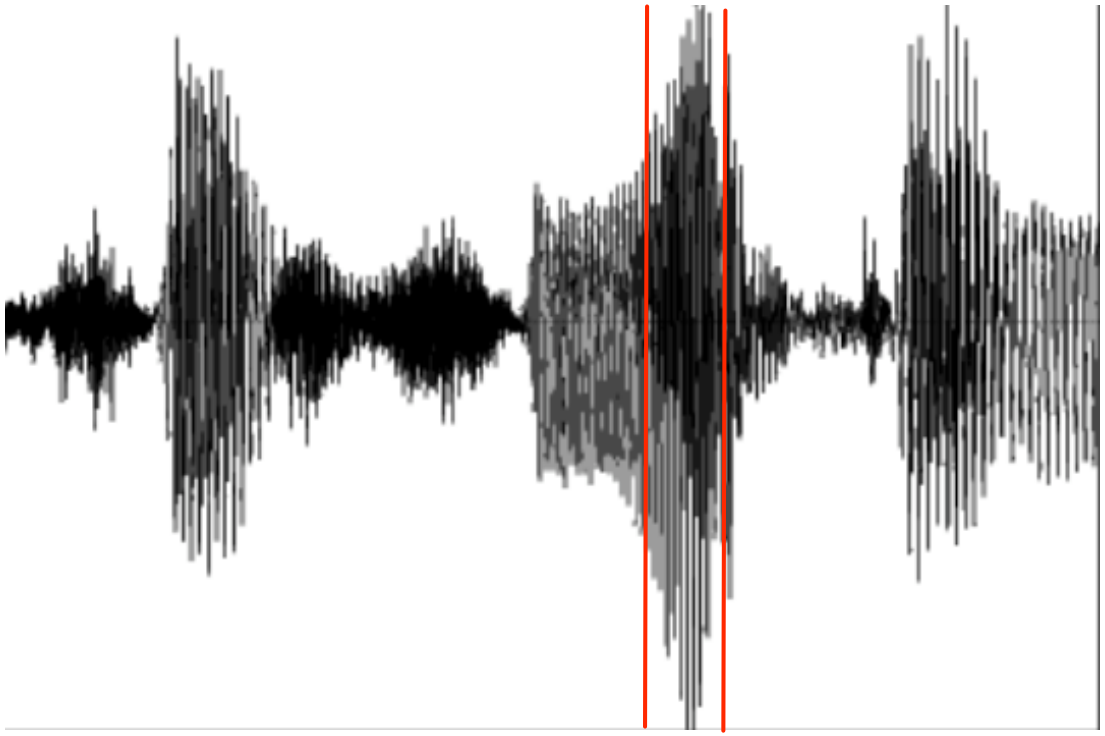
# Découpage centiseconde

- un segment toutes les 1/100 sec (échantillonnage à 16 kHz)
- les segments sont tous de la même longueur



# Observations discrètes

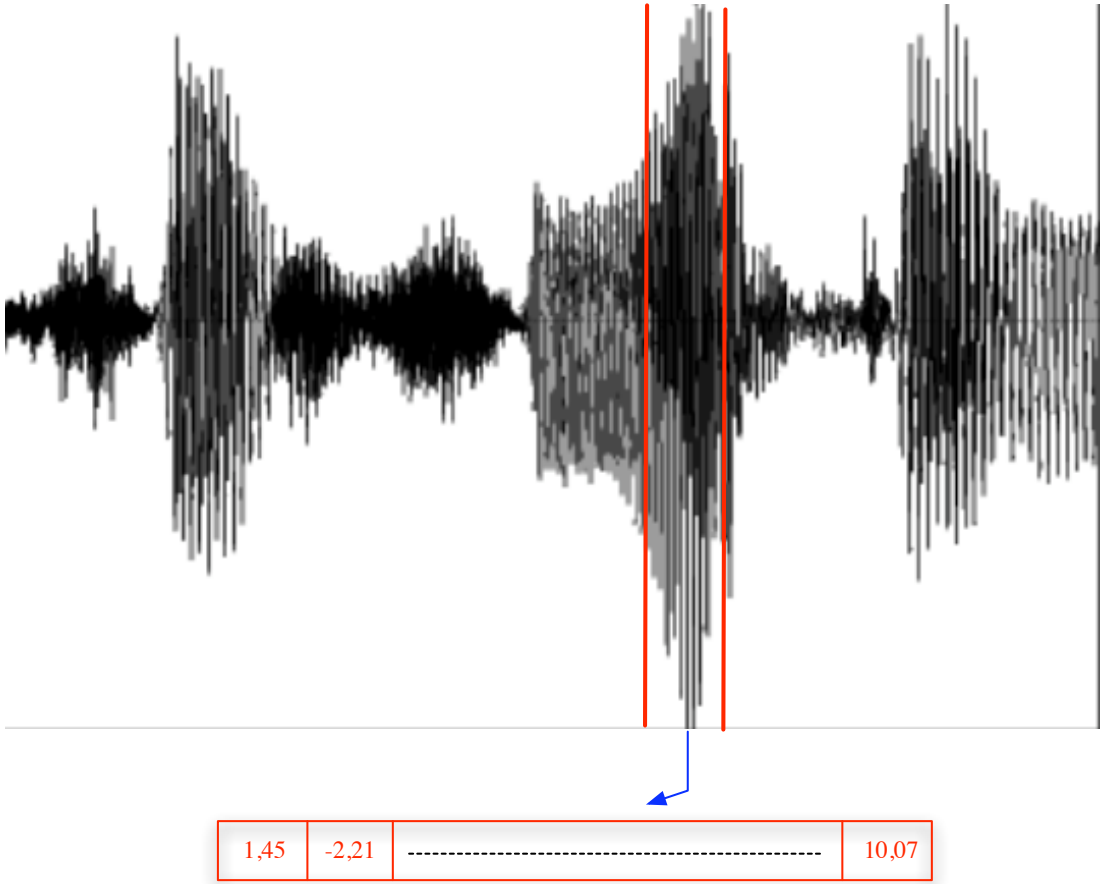
- on discrétise le segment par une étiquette
- toutes les valeurs du segment sont représentées par un symbole



[a]

# Observations continues

- le segment est représenté selon son spectre
- toutes les valeurs du segment sont représentées par un vecteur de coefficients cepstraux



# Coefficients cepstraux

- Issus d'une fonction inverse du spectre
- Principaux types (présentés par P. Deléglise)
  - MFCC (*Mel Frequency Cepstral Coefficient*)
  - LPCC (*Linear Prediction Cepstral Coefficient* )

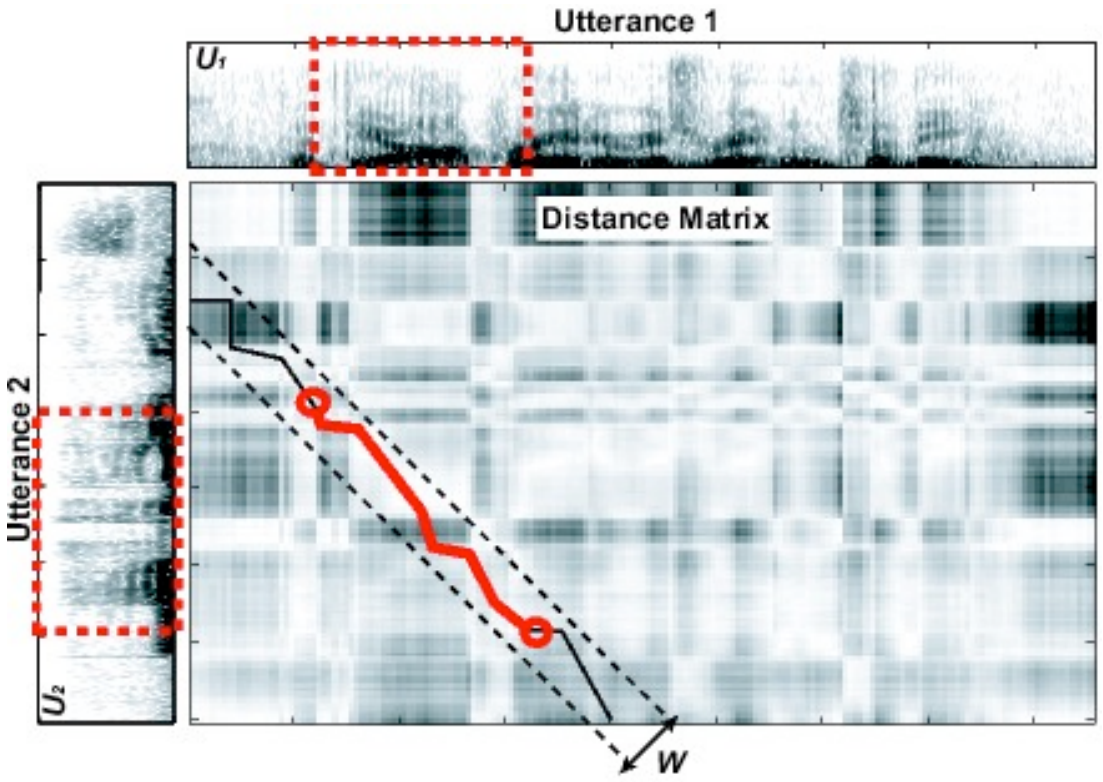
# Reconnaissance du signal

- assimilé à la reconnaissance des formes
- dans notre cas: reconnaissance de formes acoustiques
- outils statistiques:
  - la comparaison dynamique
  - les Modèles de Markov Cachés

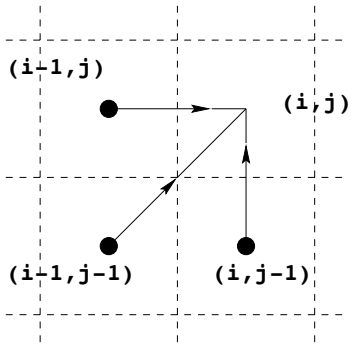
# Principe

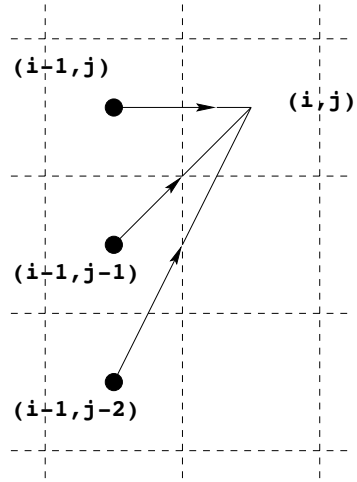
Comparaison dynamique des points:

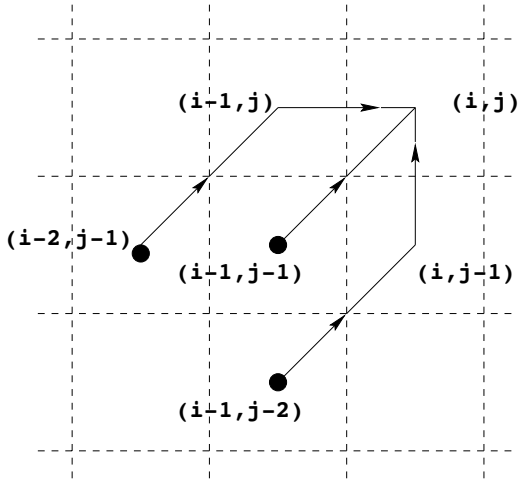
1. d'une forme de référence
2. d'une forme de test



# Contraintes de cheminement:



# Problème

## Inconvénient

- volume des formes de références $\nearrow$

# Structures

1 MMC = 3 ensembles:

# Structures

1 MMC = 3 ensembles:

- un ensemble de transitions A entre des états

# Structures

1 MMC = 3 ensembles:

- un ensemble de transitions A entre des états
- un ensemble de lois d'émissions B

# Structures

1 MMC = 3 ensembles:

- un ensemble de transitions A entre des états
- un ensemble de lois d'émissions B
- un ensemble de probabilités initiales $\Pi$

# Etats

Ensemble $\{q_1, q_2, q_3, q_4, q_5\}$

# Transitions

L'ensemble A contient les états et leurs transitions

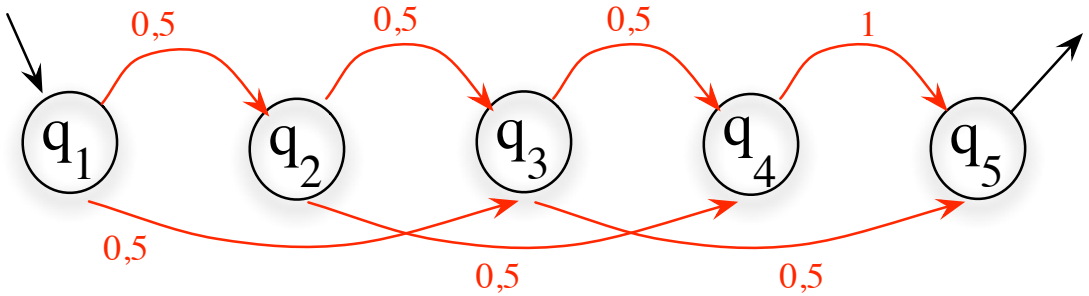
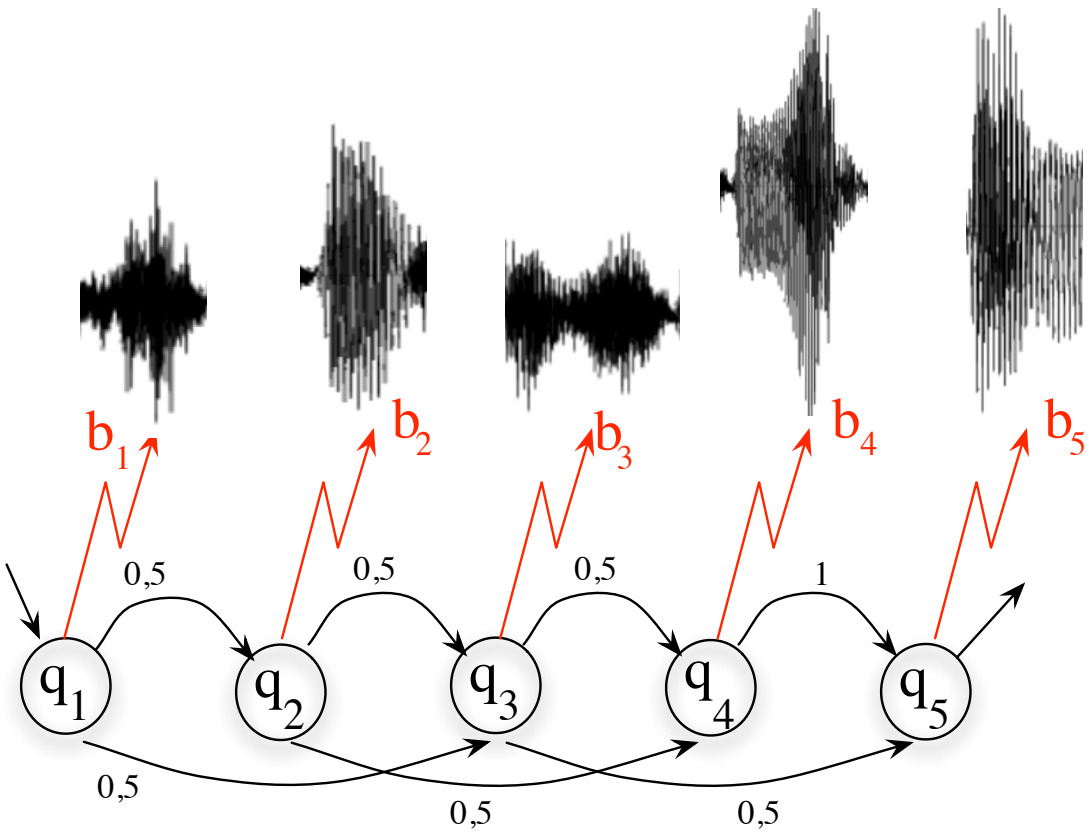



Table: Matrice A de transitions

| | q1 | q2 | q3 | q4 | q5 |
|---|---|---|---|---|---|
| q1 | 0 | 0.5 | 0.5 | 0 | 0 |
| q2 | 0 | 0 | 0.5 | 0.5 | 0 |
| q3 | 0 | 0 | 0 | 0.5 | 0.5 |
| q4 | 0 | 0 | 0 | 0 | 0.5 |
| q5 | 0 | 0 | 0 | 0 | 0 |

# Lois d'émission

- Ensemble B = $\{b_1, b_2, b_3, b_4, b_5\}$
- $b_i$ : loi gaussienne représentant une forme acoustique



# Lois d'émission

- Une loi émet la probabilité d'émission d'une observation
- dans la pratique log(proba) car problème de représentation
- Types de lois
  - lois discrètes
  - lois continues

# Lois Discrètes

- Tableau représentant les fréquences de chaque étiquette

Table: Exemple de loi discrète

| Etiq. | a | e | ei | i | ... |
|---|---|---|---|---|---|
| proba $p$ | 0.3 | 0.01 | 0.05 | 0.004 | ... |

avec :

$$\sum_{i=1}^{n} p_i = 1$$

# Calcul d'une loi Discrète

Calcul d'une loi

= proba d'émission d'une observation

= proba d'une étiquette

Pour une étiquette "a":

$$b(\text{"}a\text{"}) = log(0.3)$$

# Lois Continues

- Structure contenant les paramètres d'une loi Gaussienne
- Pour des observations de $N$ coefficients, on doit retrouver au moins:

**paramètres d'une loi continue**

- un vecteur moyenne de $N$ éléments
- un vecteur variance de $N$ éléments

- $+$ Valeurs de constantes pour calcul plus rapide dans la pratique

# Calcul d'une loi Continue

Calcul d'une loi

= proba d'émission d'une observation

= proba d'un vecteur $Y$ de $N$ coefficients: $Y = \{y_1, y_2, \ldots, y_N\}$

Pour un $y_i$:

$$\sum_{i=1}^{N}[(-0.5\ log(var_i\ 2\pi)] - \sum_{i=1}^{N}[\frac{(y_i - moy_i)^2}{2var_i} - (-0.5\ log(var_i\ 2\pi))]$$

$\Rightarrow$ On ajoute en général les parties indépendantes de $y_i$

## paramètres d'une loi continue

- moy[$N$]
- var[$N$]
- $\sum_{i=1}^{N}[(-0.5\ log(var_i\ 2\pi)]$
- $-0.5\ log(var_i\ 2\pi)$ pour tous les $i$

# Probabilités Initiales

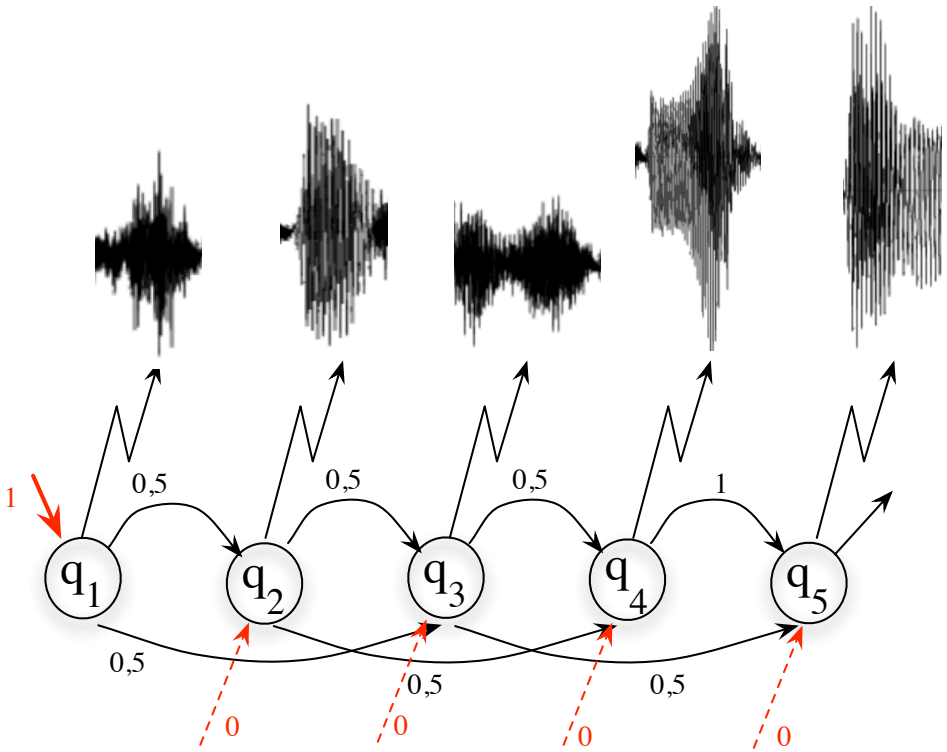



Table: Matrice Π

| q1 | q2 | q3 | q4 | q5 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |

# Apprentissage des MMC

## A partir d'un ensemble d'apprentissage

ensemble d'observations avec leur transcription

## 2 méthodes principales

- par l'algorithme de Viterbi
- par l'algorithme de Baum-Welch

# Généralités

## Algorithme de Viterbi

recherche le meilleur chemin au sens probabiliste,
ayant généré la suite d'observations $Y = \{y_1, ....., y_T\}$

# Passe avant

1. Construction de la table Q
   $Q(t, q_j)$ probabilité d'émission
   - de la sous suite d'observations $y_1 ... y_t$
   - le long du chemin le pus probable formé de $t$ états

$$\begin{cases} Q(t, q_j) = \max\ Q(t-1, q_i)\ a_{ij}\ b_j(y_t) & \text{si } t > 1 \\ Q(1, q_j) = \pi_j\ b_j(y_1) \end{cases}$$

2. Calcul vraisemblance suite d'observations / MMC:

$$\text{score MMC} = \max\ Q(T, q_j) \text{ avec } q_j \in \text{états finals}$$

# Passe arrière

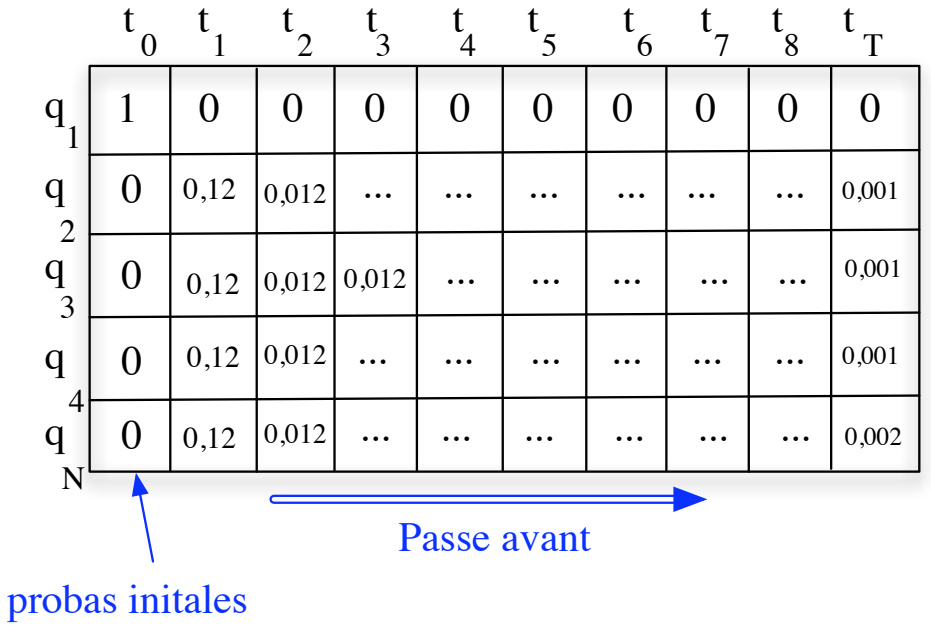
1. Recherche du meilleur chemin avec les pointeurs arrières sauvegardés dans une table $\Psi$

$$\begin{cases} \Psi(T, q_j) = \arg\max \text{ score MMC} \\ \Psi(t, q_j) = \arg\max\ Q(t-1, q_i)\ a_{ij} \end{cases}$$

2. Ré-estimation des paramètres des états $\in$ meilleur chemin
   - probabilités des transitions
   - paramètres des lois d'émission

# Synopsis 1° passe

Affectation de la table Q :



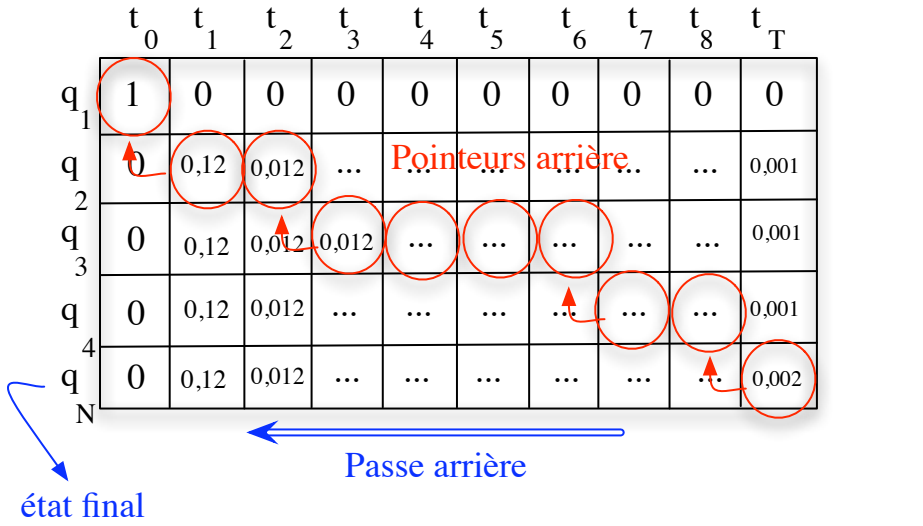
# Algo 1° passe

Peut se résumer à 3 boucles imbriquées:

```
max = - infini ;
for(t=1 ; t<=T ; t++)
    for(i=0 ; i<NB_ETATS ; i++)
    {
        for(j=0; j<NB_ETATS ; j++)
        {
            w = Q[t,j] a[j,i] b[i](y[t]) ;
            if( max < w ) max = w ;
        }
        Q[t+1,i] = max ;
    }
```

# Synopsis 2° passe

Utilisation des pointeurs arrières :




séquence d'états du chemin = $q_1\ q_2\ q_2\ q_3\ q_3\ q_3\ q_3\ q_4\ q_4\ q_N$

# Généralités

## Algorithme de Baum-Welch

recherche de tous les chemins
ayant générés la suite d'observations Y= y1,.....,yT

# Généralités

Repose sur le calcul de deux fonctions :

la fonction forward $\alpha(t, q_j)$ représente

- la probabilité d'observer les t premières observations
- et d'être à l'instant t dans l'état $q_j$

la fonction backward $\beta(t, q_j)$ représente

- la probabilité d'observer les T-t dernières observations
- sachant que l'on est à l'instant t dans l'état $q_j$.

# Passe avant

Calcul des tables $\alpha$ et $\beta$

$$\begin{cases} \alpha(t, q_j) = \sum \alpha(t-1, q_i)\ a_{ij}\ b_j(y_t) & t > 1 \\ \alpha(1, q_j) = \pi j\ b_j(y_1) \end{cases}$$

$$\begin{cases} \beta(t, q_j) & = \sum a_{ji}\ b_i(y_{t+1})\ \beta(t+1, q_i) \text{ avec } t < T \\ \beta(T, q_j) & = 1 \text{ si } q_j \text{ état final} \\ & = 0 \text{ sinon} \end{cases}$$

# Passe avant

Calcul vraisemblance de la suite d'observations par rapport au MMC:

$$\text{score MMC} = \sum \alpha(T, q_i)$$

ou

$$\text{score MMC} = \sum \pi i \; b_i(y_1) \; \beta(1, q_i)$$

tel que $q_i$ soit un état final

# Principe

## Réestimation des paramètres

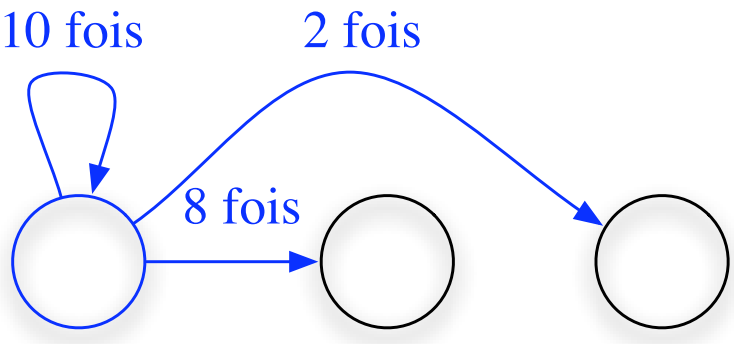
Viterbi : des états du meilleur chemin

Baum-Welch : de tous les états

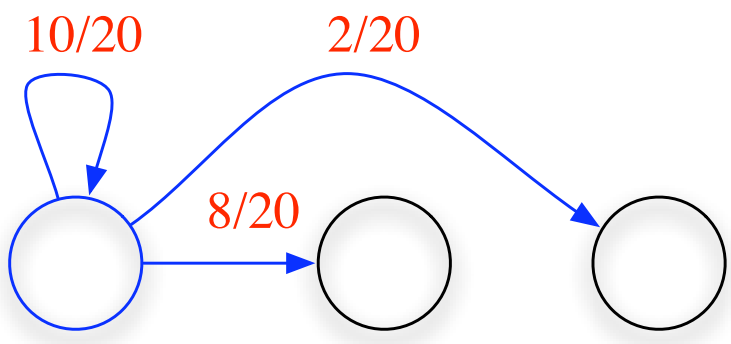
# Réestimations des transitions

Redistributions des probabilités de transitions de chaque état au bout de $N$ itérations:

1. Comptage des transitions utilisées:




2. Recalcul:



# Réestimations des Lois d'émission

2 types de lois:

- Lois discrètes
- Lois continues

# Lois discrètes

Prise en compte de la fréquence des observations discrètes qui sont émises par cette loi

1. Comptage des étiquettes vues sur cette loi:

[au] [au] [au] [o] [o] [o] [u] [u] [u]

observations

loi

| [au] | [o] | [u] |
|---|---|---|
| 0,5 | 0,3 | 0,2 |

2. Recalcul:

loi

| [au] | [o] | [u] |
|---|---|---|
| 0,3 | 0,3 | 0,3 |

# Lois continues

Prise en compte de la fréquence des observations continues qui sont émises par cette loi

1. Cumul des vecteurs de coefficients vus sur cette loi:
2. Recalcul:
   - de la moyenne
   - de la variance

# Remarque

## Apprentissage basé sur le Maximum de vraisemblance

- Si données $\nearrow$ , modélisation statistique $\nearrow$
- Si on veux avoir un bon apprentissage $\Rightarrow$ Volume de données d'apprentissage important
- Pb du manque de données

# Décodage

## Baum-Welch

Utilisé

- pour le "scratch"
- quand données ↘

## Viterbi

Utilisé par la suite

## 2 configurations abordées

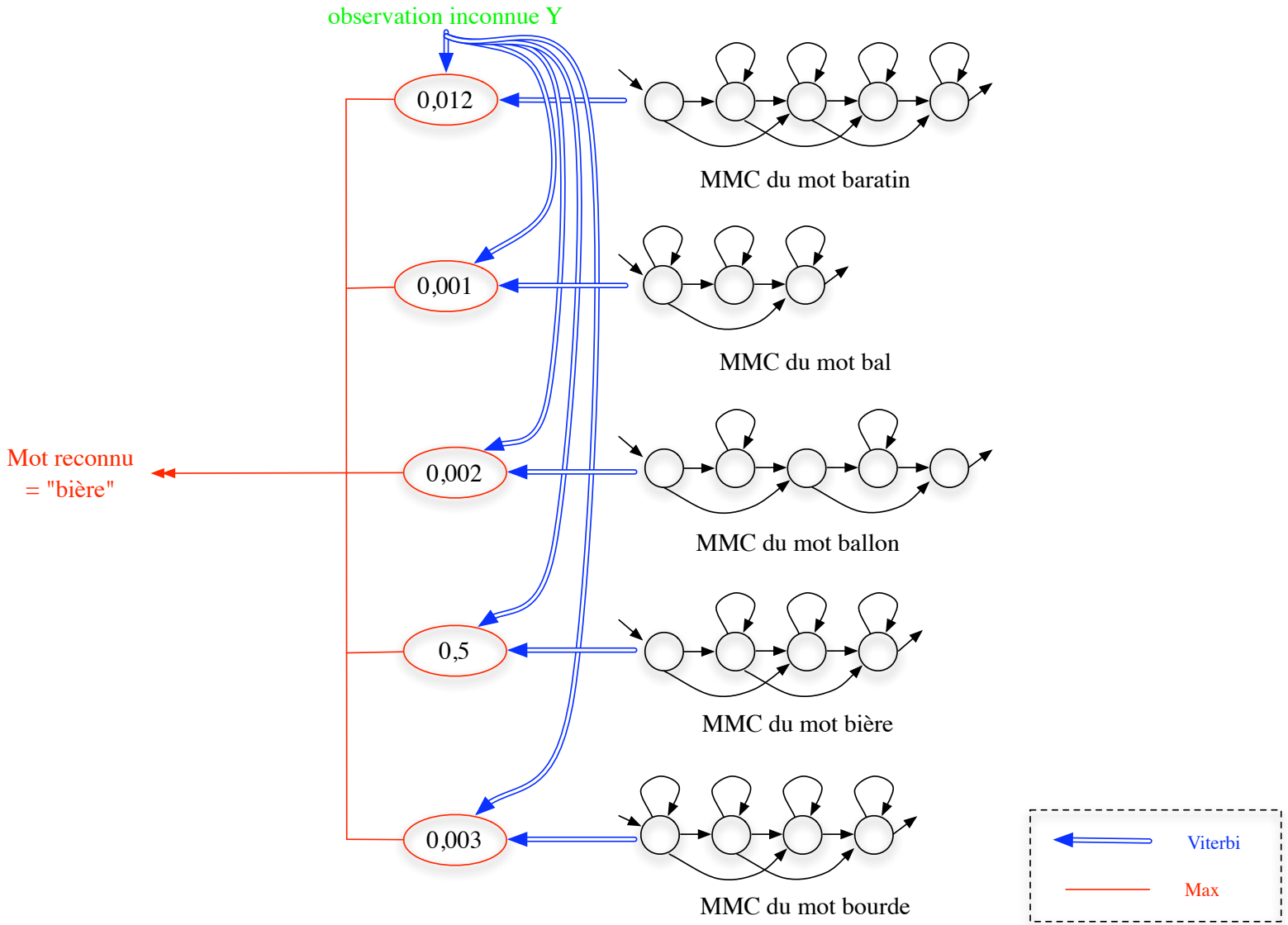
- décodage de mots isolés (1 mot à la fois)
- décodage de phrase (plusieurs mots à la fois)

# Topologie du système

## Système de RAP en mots isolés

- Chaque mot est représenté par un MMC
- Une observation $Y = \{y_1, ..., y_T\} = 1$ mot
- Calcul du score de chaque MMC pour $Y$
- Max score $\Rightarrow$ 1 MMC $\Rightarrow$ mot reconnu

# 1 MMC/mot


observation inconnue Y
0,012
MMC du mot baratin
0,001
MMC du mot bal
Mot reconnu
= "bière"
0,002
MMC du mot ballon
0,5
MMC du mot bière
0,003
MMC du mot bourde
Viterbi
Max

# Evaluation du système

- Sur un ensemble de test $\neq$ ensemble d'apprentissage
- Comparaison du résultat du décodage avec une référence
  Exemple avec test de 3 observations:

| | $Y_1$ | $Y_2$ | $Y_3$ |
|---|---|---|---|
| REF | ballon | bière | bal |
| HYP | baratin | bière | bol |

- Taux de reconnaissance en mots reconnus
  *% de reco*: 0,33% ou *% d'erreur*: 0,66%

$\Rightarrow$ On cherche donc à maximiser le taux de reco
(ou à minimiser celui d'erreur)

# Limites

## Inconvénients

- Vocabulaire fermé
- Si vocabulaire $\nearrow$ stockage des MMC $\nearrow$

# Cadre du système

## Système de RAP en dictée vocale

Dictée vocale

= parole continue

= grand vocabulaire (>100K mots)

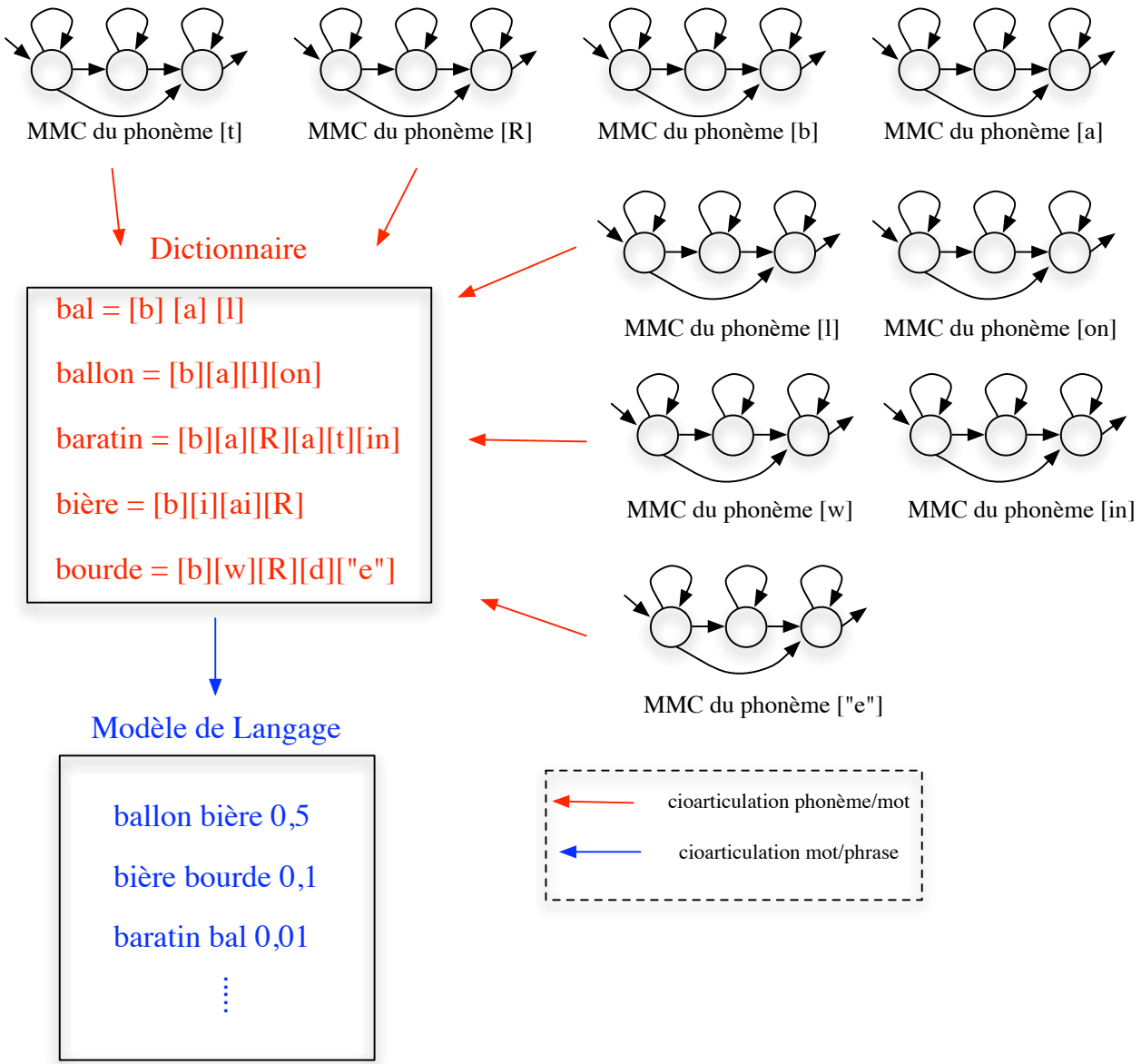
= reconnaissance de phrases (+sieurs mots qui se suivent)

- Si vocabulaire de $N$ mots alors théoriquement
  1 MCC $\Rightarrow$ toutes les phrases possibles avec $N \times N$ mots
- Si vocabulaire $\nearrow$ alors impossible de modéliser toutes les combinaisons de mots
- Eclatement de la modélisation selon le modèle de la parole en 3 niveaux

# 3 niveaux de Modélisation

1. Niveau acoustique
   - Modélisation de chaque son que l'on peut produire
   - Modélisation par MMC
   - En général 1 MMC / phonème ($\approx$ 36 pour le français)
2. Niveau lexical
   - Modélisation des mots
   - Articulation des sons en mots
   - Modélisation par Dictionnaire (correspondance entre un mot et une suite de phonèmes)
3. Niveau syntaxique
   - Modélisation des phrases
   - Articulation des mots en phrases
   - Modélisation par Modèle de Langage (probabilité que des mots se suivent)

# Les 3 niveaux de la parole



# Composants

- Modèles acoustiques
  - MMC en contexte gauche et/ou droit, triphone, diphone,...
  - Modèles segmentaux: *loi d'émission sur N segments*
  - Modèles multibandes: *1 MMC/bande de fréquence + recombinaison*
  - Réseaux bayésiens: *généralisation des dépendances entre les variables de la modélisation*
    (dans un MMC $q_i$ à $t \longrightarrow$ seulement de $q_j$ à $t-1$ et de $y_t$)

- Dictionnaire
  - avec variantes de prononciation

# Composants

- Modèle de Langage (développés par Y. Estève)
  - modèle statistique: *bigramme, trigramme,. . . ,n-grammes*
  - modèles à base de classes de mots
  - modèles cache: *stockage et ↗ proba mots déjà rencontrés*
  - modèles distants: *relie les mots distants dans une phrase*
  - triggers: *modélise le fait qu'un mot peut en déclencher un autre*
  - modèles à horizon variable: *choisi le n-gramme le plus approprié pour calculer la proba d'un mot*
  - modèles à base de séquences: *traite certaines séquences de mots comme un seul mot*

# Graphe de décodage

Actuellement: décodage lexical et acoustique en parallèle

Modèle à plat ($N$ MMC-mots reliés entre eux) trop complexe
$\Rightarrow$ Graphe de décodage ou Arbre lexical

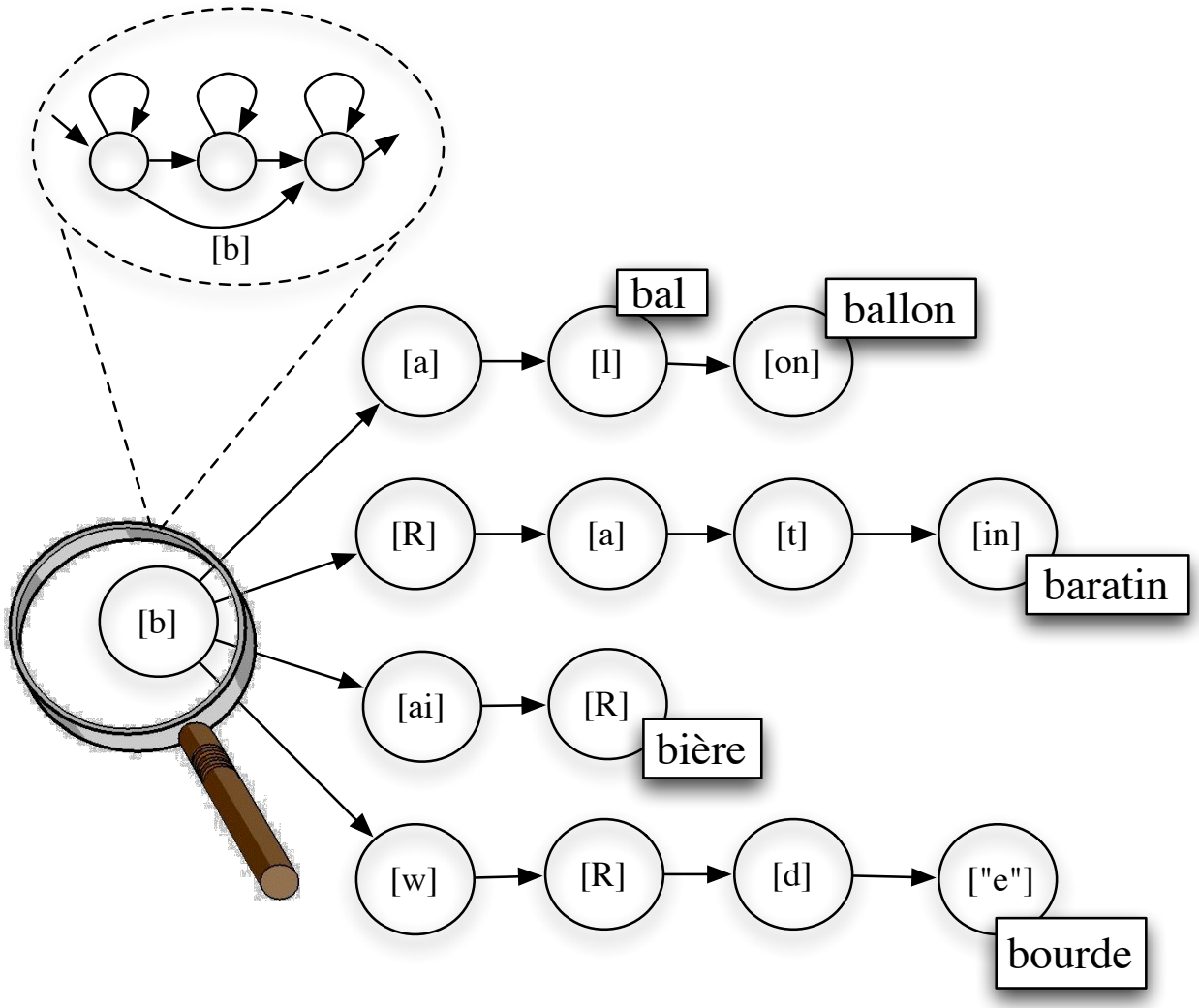
- Compilation des niveaux lexical et acoustique
- Représente le décodage d'un mot
- Organisé en arbre

# Graphe de décodage

Tous les mots sont représentés par un seul MMC (graphe de décodage)

- état initial = racine de tous les mots
- état final = terminaison d'un mot

# Graphe de décodage



# Recherche des solutions

- Une observation $Y = \{y_1, ..., y_T\}$ = segments centisecondes correspondant à $N$ mots
- Quand une observation $y_t$ est alignée sur un état final $\Rightarrow$ décodage d'un mot
- Il faut donc "Reboucler" à la racine pour décoder les mots suivants

Une recherche exhaustive est impossible (on ne peut stocker ou calculer tous les éléments de la table $Q[N, T]$)

- $N$ de l'ordre de $5 \times 100.000$
- $T$ de l'ordre de 1000

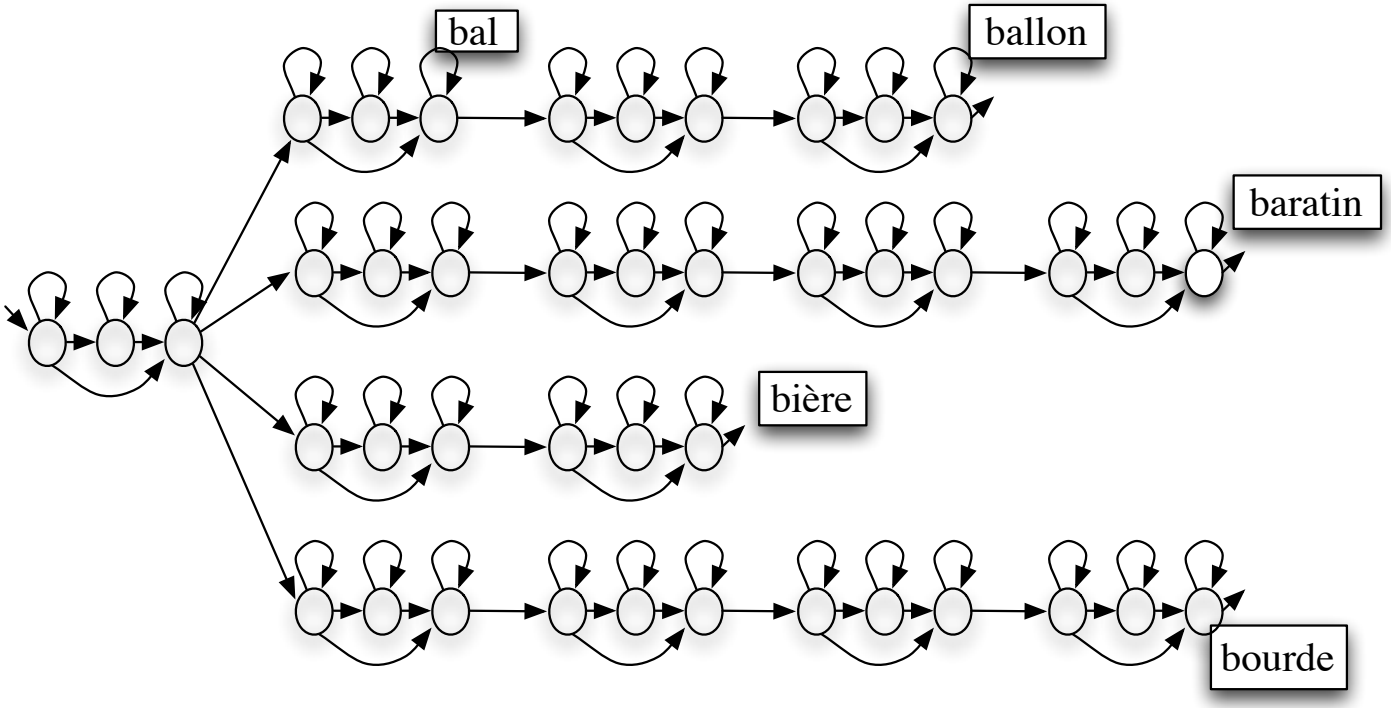
$\approx$ 8∅ $\times$ 500 *Mega* $\approx$ 4 *Giga*

# Algorithme du jeton

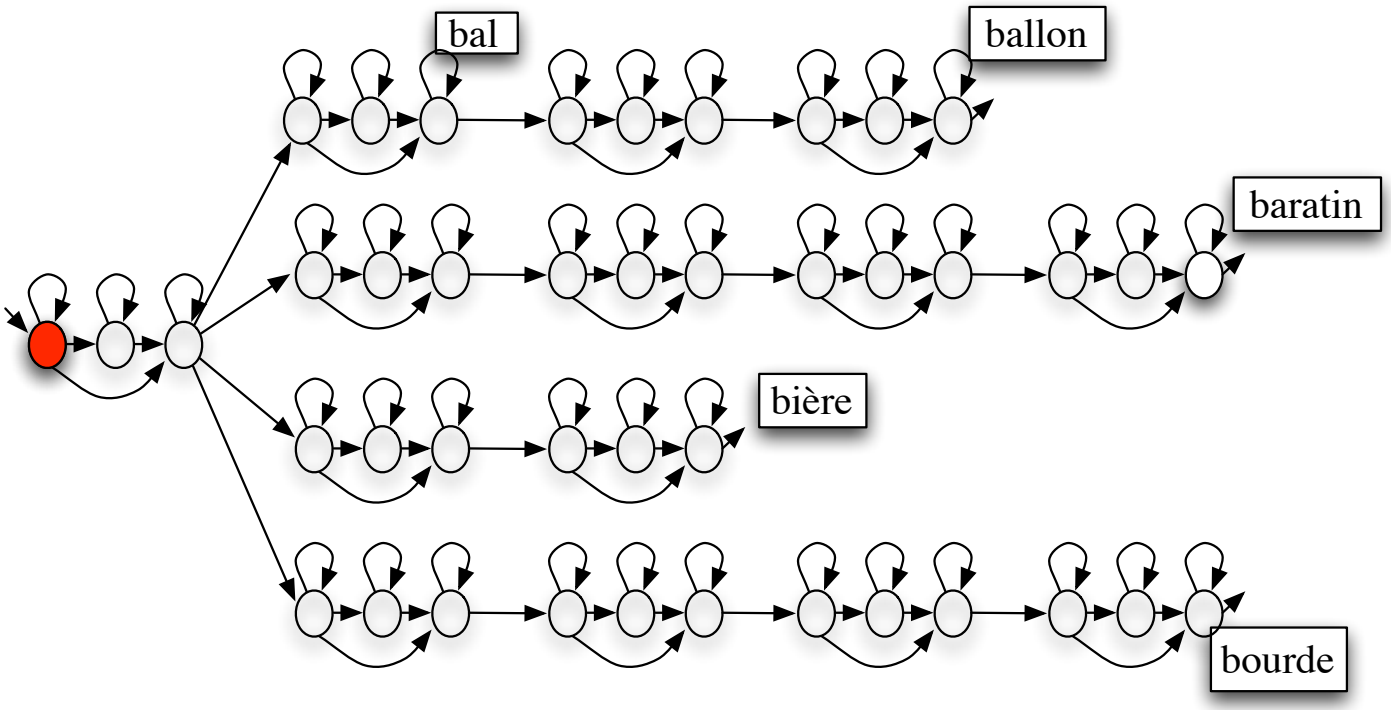
Arbre lexicographique $\Rightarrow$ Algorithme du passage du jeton (*token passing*) (développé par P. Deléglise)

- Viterbi dont la table Q est projetée dans l'arbre lexicographique
- Au départ: un jeton dans tous les états $q_i$ tels que $\pi_i \neq 0$
- Pour une observation $\longrightarrow$ propagation du jeton par la formule de Viterbi dans les états accessibles dans l'Arbre
- Un jeton contient le score de vraisemblance + l'historique (mots rencontrés dans l'Arbre)
- Phrase décodée = jeton de score max à $y_T$

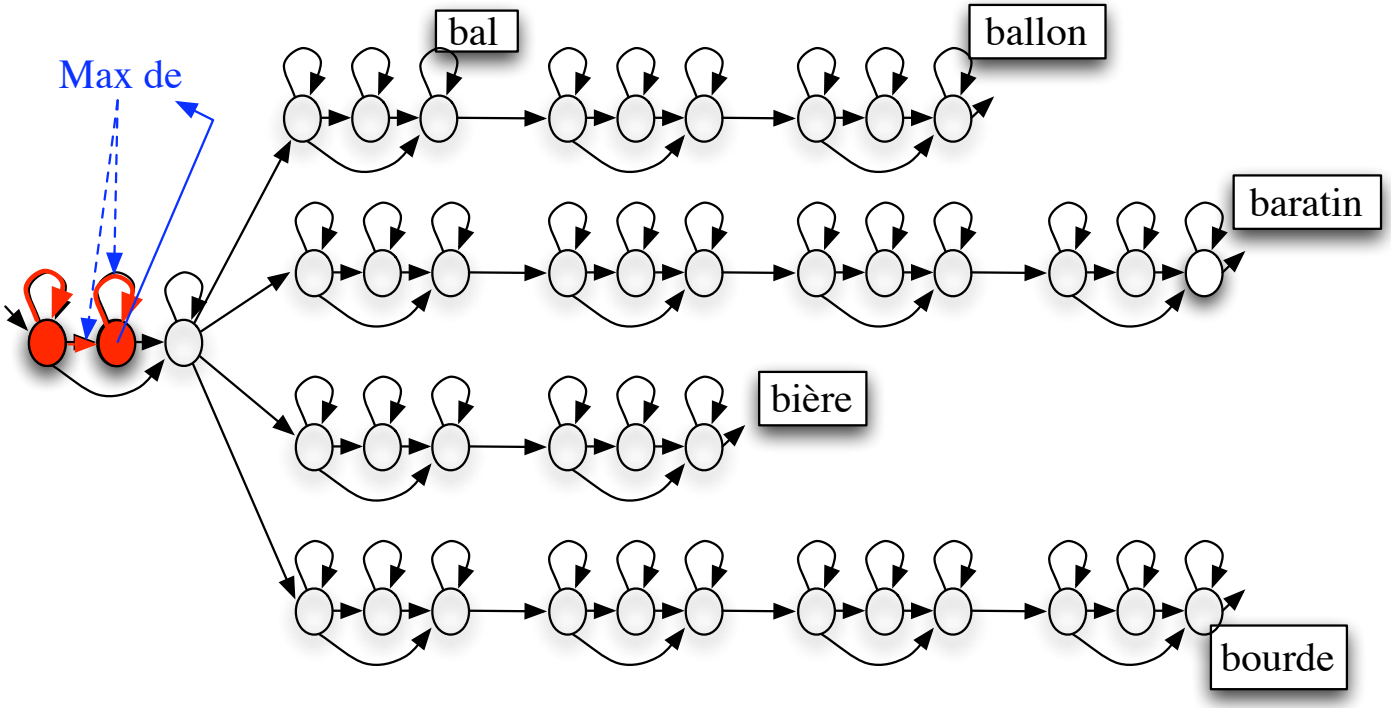
# Algorithme du jeton



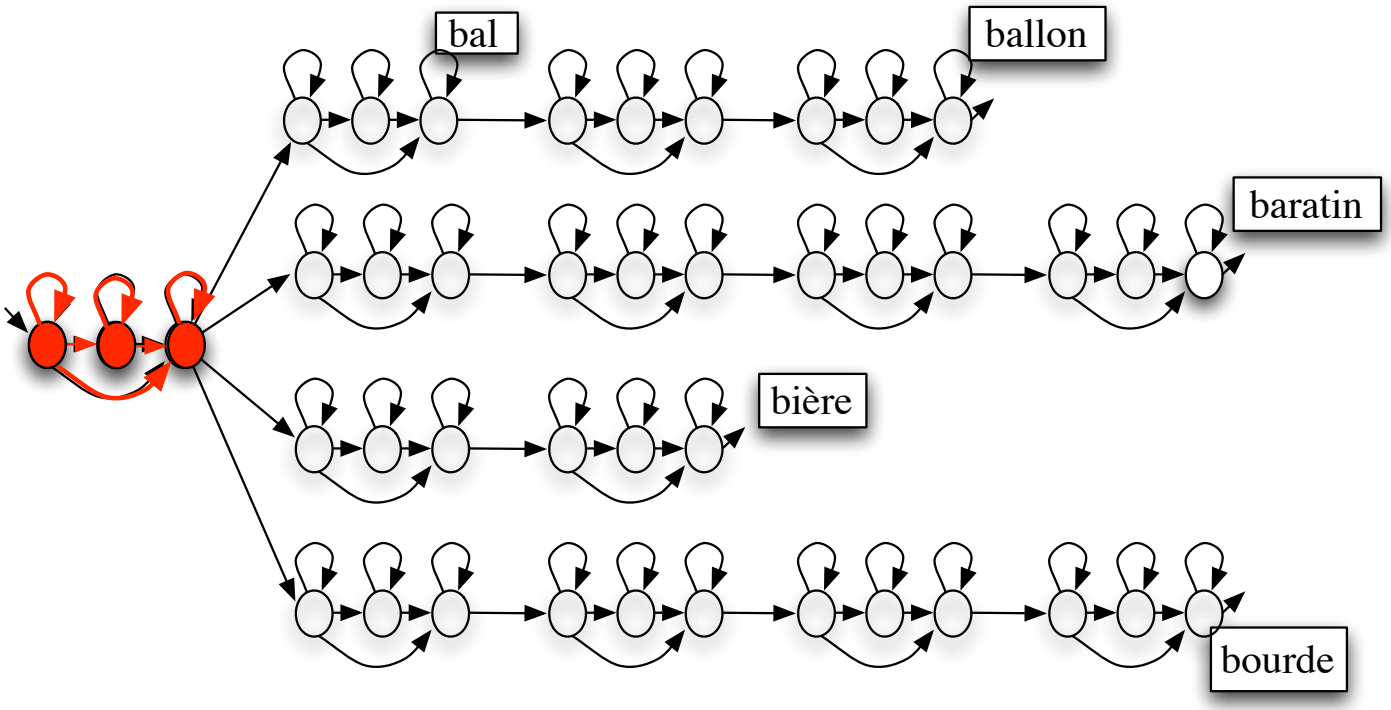
# Algorithme du jeton



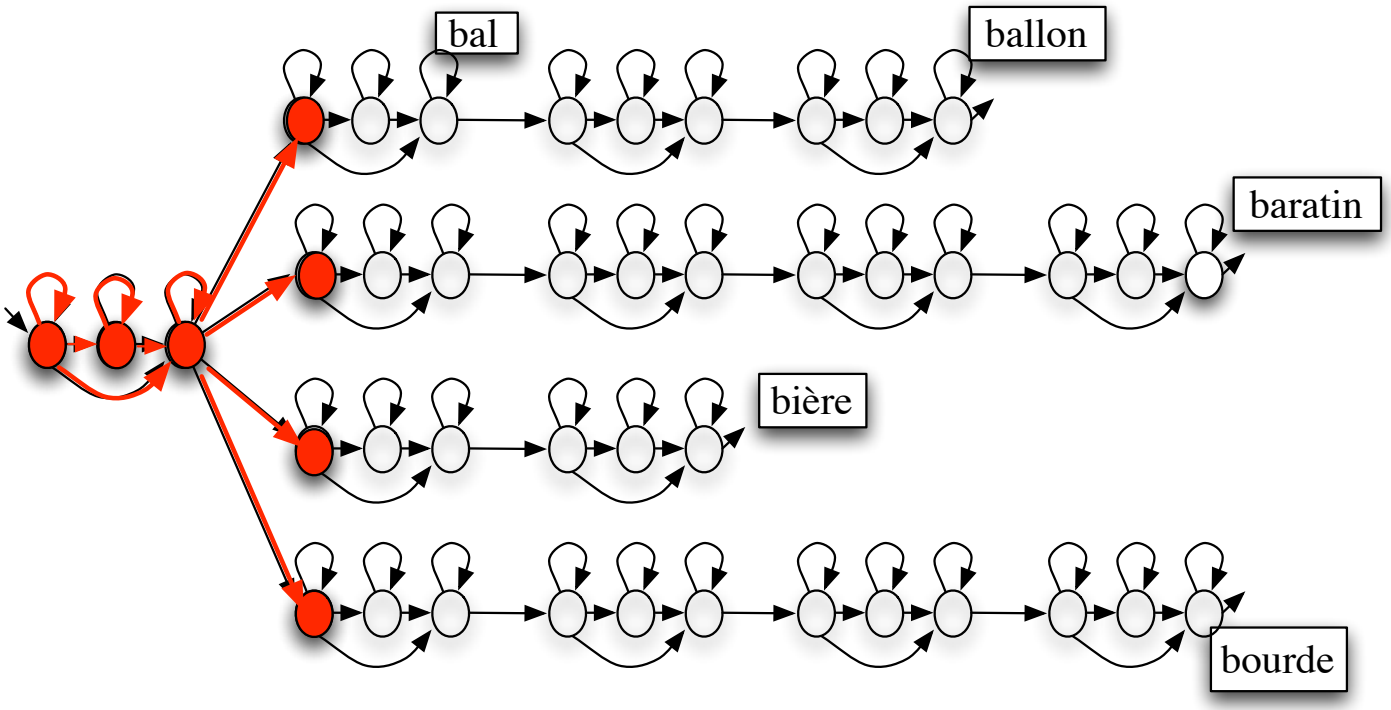
# Algorithme du jeton



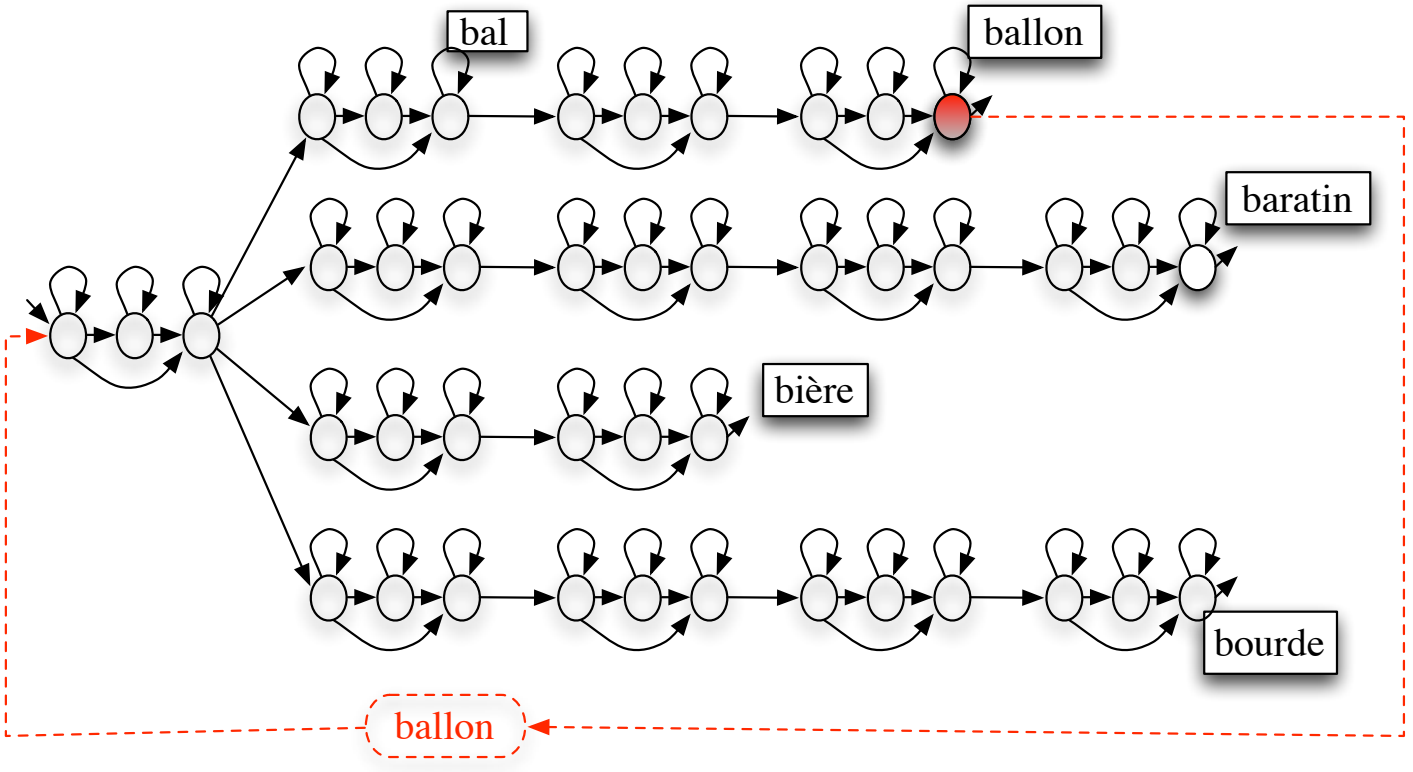
# Algorithme du jeton



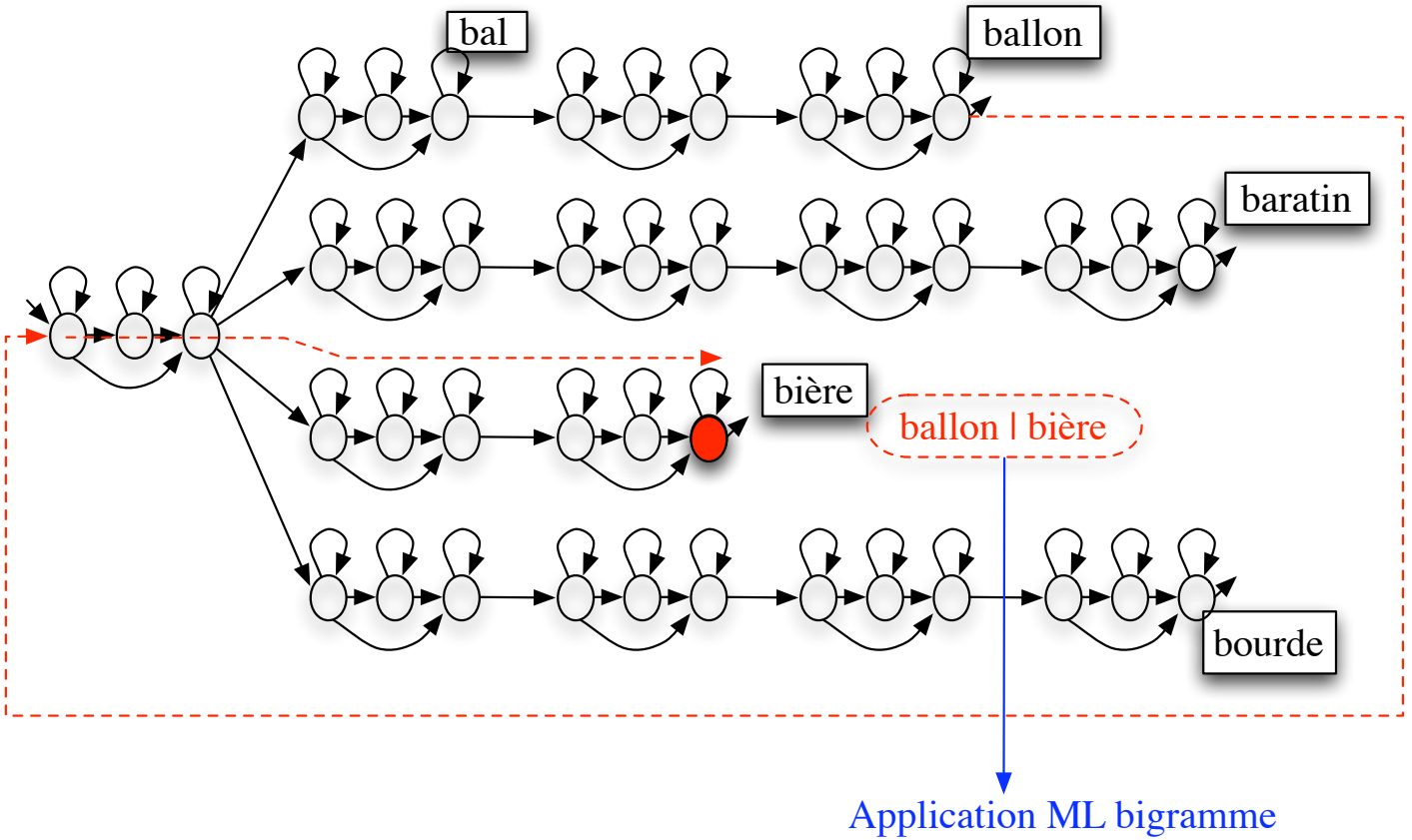
# Algorithme du jeton



# Algorithme du jeton



# Algorithme du jeton



# Faisabilité de la recherche

Recherche exhaustive de toutes les hypothèses pour $y_t$ impossible
Techniques :

→ Elagage (*Beam-Search*): faire des faisceaux dans l'espace de recherche
Quelques méthodes d'élagage:

- % par rapport au score du meilleur chemin
- Nombre maximum d'hypothèses

→ Appariement rapide (*Fast match*): Estimation des phonèmes/mots qui ont pu être prononcés pour mieux paramétrer l'élagage

→ Calcul des lois: ne calculer que celles qui sont significatives (pas trop éloigné de $y_t$), cache de celles déjà calculées

# N meilleures solutions

- N meilleurs chemins
- au lieu d'un seul jeton, on stocke $N$ jetons dans un état
- chaque jeton représente un chemin différent
- À la fin $\longrightarrow$ $N$ meilleurs chemins

# Plusieurs passes

1° passe :

- Arbre lexicographique
- Viterbi avec jeton
- ML "simple" bigramme

$\longrightarrow$ graphe ou treillis des meilleures solutions

$\Downarrow$

2° passe :

- algorithme A*
- ML "complexe" trigramme ou $\nearrow$

# Evaluation du système

Taux de reco en mots Corrects, Insérés, Supprimés, Substitués

- Taux d'erreur Word Error Rate

$$\mathrm{WER} = \frac{S + D + I}{N}$$

- Taux de reconnaissance de mots, Word Recognition Rate

$$\mathrm{WRR} = 1 - \mathrm{WER} = \frac{N - S - D - I}{N} = \frac{H - I}{N}$$

avec

N : nombre de mots de référence
S : nombre de substitutions (mots incorrectement reconnus)
D : nombre de suppressions (mots omis)
I : nombre d'insertions (mots ajoutés)
H : nombre de mots correctement reconnus

# Autres aspects

- Résistance au bruit, robustesse de la parole
- Au lieu de reconnaître des formes de mots $\longrightarrow$ reconnaître les caractéristiques du locuteur
  $\Rightarrow$ reco du locuteur
  $\Rightarrow$ segmentation d'une phrase en locuteur
  (Présentation par S. Meignier)
- transcription d'émission radiophonique, campagnes ESTER
  *Évaluation des Systèmes de Transcription enrichie d'Émissions Radiophoniques*, (équipe "parole" du LIUM)

# Conclusion

- Difficultés de la RAP
  - variabilité de la voix
  - difficile de modéliser le processus d'apprentissage/reconnaissance humain
- Contraintes d'un système
  - ce que l'on peut en attendre, avec son degré d'incertitude
  - volume de données d'apprentissage
- Limites d'un système
  - restriction dans le vocabulaire
  - restriction dans les locuteurs
- Questions ?