Reconnaissance de la parole

Sommaire

Un système de reconnaissance de la parole vise à transformer un signal acoustique reçu en entrée en une séquence de mots la plus proche possible de celle prononcée par l'utilisateur.
Soit $X = x_1, ..., x_T$ une séquence d'observations acoustiques représentant le signal de parole ; le système de reconnaissance de la parole recherche la séquence de mots la plus vraisemblable par rapport au signal acoustique en entrée. Il s'agit donc de trouver la séquence de mots $\hat{W}$ qui maximise la probabilité *a posteriori* $P(W|X)$ de la séquence de mots sachant le signal acoustique. Ceci revient à résoudre l'équation suivante :

$$\hat{W} = \underset{W}{argmax} P(W|X) \tag{1.1}$$

L'utilisation de cette formule est très difficile à cause de l'estimation de la probabilité $P(W|X)$. Cette difficulté réside dans la grande variabilité dans l'ensemble de départ des observations acoustiques. Il est plus facile d'estimer la probabilité d'avoir une certaine séquence d'observations acoustiques X sachant une séquence de mots W. La formule de Bayes permet de décomposer le terme $P(W|X)$:

$$\hat{W} = \underset{W}{argmax} \frac{P(W) \cdot P(X|W)}{P(X)} \tag{1.2}$$

Le problème est réduit à un problème d'optimisation par rapport à la séquence de mots W. La probabilité de la séquence d'observations acoustiques $P(X)$ ne dépend pas de la séquence de mots W ce qui ramène le problème d'optimisation à :

$$\hat{W} = \underset{W}{argmax} P(W) \cdot P(X|W) \tag{1.3}$$

Un système de reconnaissance de la parole a pour but de trouver la séquence de mots la plus vraisemblable par rapport au message prononcé par le locuteur. Pour ce faire, le système utilise différents modules et modèles pour analyser et décoder le signal acoustique reçu.

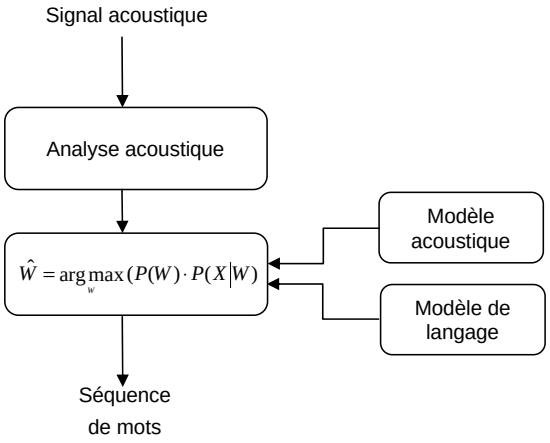

FIGURE *1.1* – Système de reconnaissance de la parole

Un système de reconnaissance de la parole (cf. figure 1.1) est composé d'un module d'analyse acoustique, présenté dans la section 1.1, suivi d'un décodeur de parole qui, à l'aide du modèle acoustique, présenté dans la section 1.2, et du modèle de langage, présenté dans la section 1.3, doit trouver une solution pour le problème d'optimisation de l'équation 1.3. La problématique de la combinaison des modèles de langage et acoustiques est traitée dans la section 1.4. L'espace de recherche utilisé par le décodeur et les différentes sorties possibles sont présentés dans la section 1.5 et les méthodes d'évaluation d'un système de reconnaissance sont détaillées dans la section 1.6.

1.1 Analyse acoustique

Le module d'analyse acoustique transforme le signal de parole en une séquence de vecteurs de coefficients qui est fournie en entrée du décodeur de parole. Les vecteurs de coefficients sont censés éliminer toute information qui n'est pas importante pour la reconnaissance comme les caractéristiques du locuteur (homme, femme), la réponse fréquentielle du canal ou toute sorte de bruit.
Le signal de parole est divisé en fenêtres temporelles (trames) et chaque portion du signal est alors analysée ; un vecteur de coefficients est le résultat de l'analyse de chaque portion. Afin de compenser dans une certaine mesure la forte pente spectrale du signal de parole dans le cas des voyelles et pour augmenter l'énergie du signal dans les hautes fréquences (spécialement pour les consonnes), le signal passe tout d'abord par un filtre de pré-accentuation de premier ordre de type passe-haut. Pour diviser les signal de parole en trames nous utilisons des fenêtres de Hanning (des fenêtres de Hamming peuvent aussi être utilisées). La longueur de la fenêtre se situe habituellement dans un intervalle de 10-32 ms et les fenêtres successives ont un facteur de recouvrement de 40-60%. Si ce recouvrement n'existait pas on perdrait les informations contenues aux bords des fenêtres. Dans nos travaux, la longueur de la fenêtre de Hanning utilisée est de 32 ms et le recouvrement de 16ms (50%) ce qui donne un vecteur de 256 points avec un recouvrement de 128 points (pour un signal échantillonné à 8kHz).
Il existe plusieurs méthodes pour analyser un signal de parole. Dans les année 70 la méthode la plus populaire était l'analyse linéaire prédictive (*linear predictive analysis*) (Makhoul, 1975) et les coefficients cepstraux LPCC (*Linear Prediction Cepstrum Coefficient*) associés (Rabiner et Juang, 1993). A partir du milieu des années 80, la représentation standard du signal de parole utilisée repose sur les *Mel-Frequency Cepstral Coefficients* (MFCC) (Davis et Mermelstein, 1980). Dans nos travaux nous utilisons les MFCC et le processus de calcul de ces coefficients est décrit ci-dessous.
Le spectre du signal de parole est calculé pour chaque trame à l'aide de la FFT (la densité spectrale est égale au carré du module de la transformé de Fourrier) ; ensuite le spectre est filtré à l'aide d'une série de filtres triangulaires passe-bande espacés de manière égale sur l'échelle de Mel. La liaison entre cette échelle et l'échelle linéaire utilisé habituellement (Hz) est donnée par l'équation : $f_{Mel} = 2595 log_{10}(1 + f_{Hz}/700)$. Dans l'échelle linéaire (Hz), les filtres sont espacés de manière linéaire jusqu'à 1kHz et ensuite la fréquence centrale et la bande passante augmentent de façon logarithmique. Le logarithme de l'énergie du signal dans chaque filtre est ensuite calculé et suite à une transformé en cosinus discret (DCT) de ces logarithmes on obtient les MFCC. La formule de calcul des MFCC à partir des logarithmes de l'énergie du signal E_k est la suivante (Davis et Mermelstein, 1980) :

$$MFCC_i = \sum_{k=1}^{N} E_k cos(i\pi \frac{k - \frac{1}{2}}{N}) \tag{1.4}$$

où N est le nombre de filtres (habituellement entre 20 et 25, nous utilisons 24 filtres) et i varie de $1, \ldots, M$, avec M étant le nombre de coefficients calculés.
La DCT a la propriété de rendre les coefficients quasiment non-corrélés. Il est alors possible de ne retenir que quelques termes de la transformé en cosinus pour représenter la

variation des logarithmes de l'énergie. Le résultat est une représentation compacte du signal de parole où seulement 8 valeurs suffisent ; le coefficient d'ordre 0 qui est lié à l'énergie de la trame est souvent remplacé par une autre mesure d'énergie. Les paramètres ainsi obtenus forment ce qu'on appel le vecteur statique.
Du fait du recouvrement des trames, les vecteur de coefficients sont sensiblement corrélés alors que la modélisation acoustique utilisée part de la prémisse que les vecteurs sont indépendants. Afin de prendre en compte les corrélations temporelles qui peuvent exister entre les coefficient cepstraux et rendre ainsi mieux compte de leur dynamique on rajoute les estimés de la dérivé première et seconde (les coefficients dynamiques) du vecteur statique. Cette idée a été proposé pour la première fois dans (Furui, 1986). Les dérivés sont calculées par régression sur quelques trames adjacentes.
Le vecteur de coefficients qui décrit chaque trame du signal de parole est donc formé du vecteur statique (MFCC et l'énergie) et de leurs dérivés premières et secondes. Ainsi, chaque trame du signal est représentée par un vecteur de 27 coefficients.

1.2 Modélisation acoustique

Pour une personne, prononcer le même mot plusieurs fois ne produit pas forcément le même résultat acoustique. La durée du signal, mais aussi la variabilité de la prononciation ne sont pas forcément les mêmes. Ces différences sont encore plus accentuées entre différentes personnes. Une modélisation statistique des différentes prononciations a été proposée au milieu des années 1970. Elle n'a vraiment été adoptée que dans la décennie suivante. Les modèles de Markov cachés (*Hidden Markov Models* - HMM) sont maintenant largement utilisés dans la modélisation acoustique parce qu'ils peuvent aisément prendre en compte des durées et de prononciations différentes pour un même mot. Leur utilisation s'est étendu) d'autres domaines, par exemple la reconnaissance des séquences du génome ou de l'écriture manuscrite. Par ailleurs, les modèles de Markov cachés et leurs applications ont été largement décrits, par exemple dans (Jouvet, 1988; Rabiner, 1989).
La modélisation acoustique d'une séquence de mots W est obtenue par une décomposition successive de celle-ci. Elle est ainsi séparée en mots qui sont à leur tour séparés en unités phonétiques (phonèmes, triphones, diphones, allophones). Chaque unité phonétique est représentée par un HMM (Rabiner et Juang, 1993), et la reconnaissance d'un signal de parole consiste à choisir la séquence de modèles qui attribue la vraisemblance la plus forte au signal observé.
Un HMM est un automate à N états $s_1, \ldots, s_N$ inter-connectés par de transitions, avec une probabilité attaché à chaque transition. Un HMM peut être vu comme un modèle où, à chaque trame, un symbole (une observation) est produit (émis) avec une probabilité donnée. Pendant le décodage, le processus émet un symbole avec une probabilité donnée. L'émission des observations peut être attachée soit aux états soit aux transitions. Ces deux points de vue sont équivalents (Jelinek, 1997), mais dans la littérature l'émission est souvent attachée aux états. Dans le cadre de nos travaux, l'émission est attachée aux transitions ce qui est plus adapté à l'utilisation des FSN (*Finite State Network*) dans le processus de reconnaissance (les entrées sont les vecteurs de coefficients

et les sorties, des séquences de mots).
La topologie principalement employée dans la littérature est un modèle gauche-droit d'ordre 1, dit de Bakis. Ceci implique donc que la probabilité de passer dans un état j dépend uniquement de l'état précédent. Le modèle est aussi supposé stationnaire, les probabilités de transition d'un état à l'autre ne dependent pas du temps. On a :

$$P(s_{t+1} = j | s_t = i) = a_{ij} \tag{1.5}$$

avec a_{ij} qui vérifie la relation suivante :

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{1.6}$$

On définit aussi la probabilité d'occupation initiale des états $P(s_1 = i) = \pi_i$ qui vérifie la relation :

$$\sum_{i=1}^{N} \pi_i = 1 \tag{1.7}$$

Comme nous l'avons mentionné, l'émission des observations peut être attachée soit aux états soit aux transitions. Lorsque le vecteur d'observations est à valeurs discrètes, on utilise une vraie valeur de probabilité pour l'émission. S'il s'agit des vecteurs à valeurs continues, comme dans note cas, une densité de probabilité est utilisée qui exprime la vraisemblance entre l'observation émise et le vecteur de coefficients. La densité de probabilité le plus souvent utilisé (dans la reconnaissance vocale), employée aussi dans notre système, est un mélange de densités gaussiennes. Le mélange peut être constitué d'une fonction (mono gaussienne) et jusqu'à un nombre maximal de gaussiennes. Plus le nombre est grand, meilleure (plus détaillée) sera la modélisation, mais le problème de l'estimation d'un grand nombre de paramètres à partir d'un volume de données restreint limite le nombre de gaussiennes utilisées afin d'obtenir une bonne estimation. La densité de probabilité d'émission d'une observation dans l'état i s'écrit :

$$b_i(x_t) = \mathcal{N}(x_t, \mu_i, \Sigma_i) \tag{1.8}$$

Un HMM est donc caractérisé par le vecteur constitué des probabilités initiales π_i, la matrice des probabilités de transition a_{ij} et la matrice des densités de probabilité d'émission $b_i(x_t)$. Les séquences de transition ne sont pas accessibles, seulement les observations émises le sont ; pour cela les modèles sont dits "cachés".

La phase d'apprentissage permet d'estimer les paramètres du modèle acoustique grâce à un corpus d'apprentissage. Il s'agit d'estimer les probabilités des transitions ainsi que les paramètres des densités d'observation associées aux états (les vecteurs de moyennes et les matrices de covariance d'un ensemble de gaussiennes). Généralement, les algorithmes EM (*Expectation-Maximization*), introduit dans (Dempster et al., 1977), et Baum-Welch (Rabiner et Juang, 1993), sont utilisés pour réaliser l'apprentissage des HMM. L'apprentissage des modèles acoustiques nécessite un volume de données assez important, la transcription du signal de parole étant aussi nécessaire. La première étape

consiste en la phonétisation de la transcription. L'étape suivante consiste en l'application de l'algorithme d'apprentissage. Ceci peut se faire de manière supervisée ou non. Pour un algorithme supervisé, les mots (ou les phonèmes) sont alignés sur le signal de parole en leur faisant correspondre un nombre de trames. Ainsi, lors de l'apprentissage, on se sert de cette information afin de contraindre la recherche du chemin optimal dans ce sous-modèle. Dans le cas non-supervisé tout le modèle est accessible. Dans la pratique, l'apprentissage des chaînes de Markov est presque toujours supervisé.
Afin d'assurer la robustesse des modèles acoustiques, les corpus d'apprentissage doivent contenir une quantité importante d'énoncés. Il est aussi important que les énoncés proviennent de plusieurs locuteurs afin de mieux modéliser la variabilité inter-locuteur. Ceci permet de construire des systèmes de reconnaissance indépendants du locuteur.

1.3 Modélisation statistique du langage

Les modèles de langage sont utilisés dans un système de reconnaissance de la parole pour guider le décodage acoustique. Trouver un bon modèle de langage influence considérablement les performances d'un système de reconnaissance vocale. Un tel modèle doit être discriminant pour favoriser certaines hypothèses par rapport à d'autres, tout en n'étant pas trop restrictif pour bien généraliser aux nouvelles données rencontrées lors d'un décodage.
Il existe deux approches différentes dans la construction d'un modèle de langage : les grammaires formelles et les modèles de langage stochastiques. Étant donné que les modèles de langage ne font pas l'objet de ces travaux nous n'allons pas détailler les grammaires formelles décrites dans (Chomsky, 1957, 1965; Pullum et Gazdar, 1982). Les modèles de langage stochastiques sont les plus répandus dans les applications de reconnaissance vocale, dont celle utilisée pour réaliser ces travaux. Nous allons faire une courte description de ce type de modèle de langage.
Le but d'un modèle de langage est d'attribuer une probabilité $P(W)$ à un séquence de mots $W = w_0, ..., w_N$. Sachant que les symboles de début et de fin de phrase, respectivement notés $< s >$ et $< /s >$, sont inclus dans la séquence de mots, la probabilité $P(W)$ peut se décomposer de la manière suivante :

$$P(W) = P(w_0, ..., w_N) = \prod_{i=1}^{N} P(w_i | w_0, ..., w_{i-1}) \tag{1.9}$$

On défini l'historique h_i du mot w_i comme étant l'ensemble des mots le précédant dans l'énoncé, $h_i = w_0, ..., w_{i-1}$.

$$P(W) = \prod_{i=1}^{N} P(w_i | h_i) \tag{1.10}$$

Le modèle doit être capable d'estimer les différentes probabilités $P(w_i|h_i)$. Même pour une longueur moyenne des historiques et pour un lexique de taille moyenne le calcul de ces probabilités conditionnelles devient problématique. Le nombre d'historiques possibles est trop important pour avoir des estimations fiables des différentes probabilités.

Une réduction de la taille des historiques est donc nécessaire afin de pouvoir estimer correctement les probabilités $P(w_i|h_i)$.

1.3.1 Approximation par modèle n-gramme

Dans les modèles *n-gramme* l'historique est réduit aux $n-1$ derniers mots précédant w_i dans l'énoncé. La probabilité du modèle de langage $P(W)$ devient :

$$P(W) = \prod_{i=1}^{N} P(w_i|w_{i-n+1}, ..., w_{i-1}) \tag{1.11}$$

Dans la pratique, n est rarement choisi supérieur à 3 pour une première passe dans le système de reconnaissance de la parole :

- pour $n = 2$, le modèle de langage est un *bigramme* et la probabilité d'un mot ne dépend que du mot qui le précède :

$$P(w_i|w_0, ..., w_{i-1}) \approx P(w_i|w_{i-1}) \tag{1.12}$$

- pour $n = 3$, le modèle de langage est un *trigramme* et la probabilité d'un mot ne dépend que des deux mots qui le précèdent :

$$P(w_i|w_0, ..., w_{i-1}) \approx P(w_i|w_{i-2}, w_{i-1}) \tag{1.13}$$

Le modèle de langages *n-gramme* est un modèle stochastique qui n'utilise aucune connaissance d'ordre sémantique ou syntaxique. Malgré sa simplicité de modélisation, ce modèle est très répandu dans les systèmes de reconnaissance vocale car il est assez discriminant et facilement utilisable avec un faible coût en termes de calcul lors du processus de décodage. En effet, un modèle *n-gramme* peut facilement se mettre sous la forme d'un modèle à états finis stochastique et s'intégrer aisément avec les HMMs du modèle acoustique (également des modèles stochastiques à états finis) pour former un FSN (*Finite State Network*). Ce type de réseau est utilisé par notre système de reconnaissance.

1.3.2 Modèle à base de classes

Une des étapes les plus importantes dans la construction d'un modèle de langage est l'apprentissage. Cette étape nécessite une grande quantité de données d'apprentissage qui ne peuvent pas être exhaustives en ce qui concerne la variabilité et la complexité d'un langage. Une variante des modèles de langage de type *n-grammes* sont les modèles de langages à base de classes de mots (Damnati, 2000; Kobus, 2006).
Certains mots peuvent avoir un comportement similaire et il est donc possible de les regrouper en classes de mots. Ceci présente deux avantages majeurs :

- le nombre d'événements à modéliser est réduit ce qui améliore le rapport entre le nombre de paramètres à estimer et la taille des données d'apprentissage.
- une généralisation est réalisée sur les événements possibles dans le *n-gramme*. Un événement non vu au niveau mot dans le corpus d'apprentissage peut être plus facilement modélisé au niveau des classes.

Dans ce contexte un mot w_i appartient à une classe c_i. L'historique n'est plus une séquence de mots, mais une séquence de classes. Le modèle est alors construit de la façon suivante à partir des $n-1$ classes précédentes :

$$P(w_i|c_{i-n+1}\dots c_{i-1}) = P(w_i|c_i)P(c_i|c_{i-n+1}\dots c_{i-1}) \tag{1.14}$$

où $P(w_i|c_i)$ est la probabilité d'appartenance du mot w_i à la classe c_i et $P(c_i|c_{i-n+1}\dots c_{i-1})$ est la probabilité de la classe c_i connaissant le n-1 classes précédentes.
Pour un modèles *bi*-classes (n=2), la probabilité de la séquence de mots W se calcule de la manière suivante :

$$P(W) = \prod_{i=1}^{N} P(w_i|c_i)P(c_i|c_{i-1}) \tag{1.15}$$

Le modèle de langage utilisé dans nos travaux est un bi-classes dont la majorité des classes est formé d'un seul mot.

1.3.3 Imbrication des modèles

Dans un système de reconnaissance on peut parfois avoir besoin de modéliser différemment certains segments dans un énoncé, comme par exemple un numéro de téléphone. On peut utiliser différent sous-modèles de langage (*n-grammes*, grammaires) qui sont imbriqué dans le modèle général.
On prend l'exemple d'un modèle bigramme P^G qui contient un certain nombre de classes de mots et un sous-modèle de langage de type bigramme P^S. Le sous-modèle de langage est inclus dans le modèle global est son comportement est assimilé à celui d'une classe de mot notée c_S. La probabilité de la séquence de mots w_1, w_2, w_3, w_4 où la séquence w_2, w_3 est gérée par le sous-modèle se calcule de la façon suivante :

$$\begin{aligned} P^{G\ avec\ S}_{w_1,w_2,w_3,w_4} = \quad & P^G(w_1|start) \cdot P^G(c_S|w_1) \cdot \\ & P^S(w_2|start) \cdot P^S(w_3|w_2) \cdot \\ & P^S(end|w_3) \cdot P^G(w_4|c_S) \end{aligned} \tag{1.16}$$

1.3.4 Lissage

Lors de l'étape d'apprentissage [1] on estime les valeurs des paramètres d'un modèle du langage. Ces paramètres sont estimés à partir du nombre d'occurrences des événements dans le corpus d'apprentissage. Or les données dont on dispose pour l'apprentissage ne sont pas exhaustives, ainsi il existe un grand nombre d'événements jamais observés. Afin d'éviter que certains événements se voient attribuer une probabilité nulle, un lissage des probabilités est nécessaire. Le principe du lissage est justement d'attribuer une probabilité non nulle aux événements jamais observés lors de l'apprentissage en prélevant une quantité de la masse de probabilités de l'ensemble des événements observés pour la redistribuer aux événements non vus. Ce processus est guidé par une

1. La technique d'*absolute discounting* (Ney et Essen, 1991) est utilisée dans nos travaux.

contrainte de normalisation qui implique que, pour un historique h, les probabilités $P(w|h)$ somment à 1 pour tous les mots w du lexique. Les principales techniques de lissage et leur performances sont décrites dans (Chen et Goodman, 1996).

1.4 Combinaison des modèles acoustiques et des modèles de langage

Comme le suggère la formule 1.3, les scores donnés par le modèle acoustique et le modèle de langage se combinent par une simple opération de multiplication. Dans la pratique, cela n'est pas aussi simple car l'ordre de grandeur des scores est très différent. En effet, la vraisemblance d'une densité de probabilité donnée par le modèle acoustique est beaucoup plus petite que la probabilité donnée par le modèle de langage : $P(X|W) << P(W)$. Les multiplier directement reviendrait à négliger l'influence d'un modèle par rapport à l'autre.
La solution la plus couramment utilisée est d'introduire un facteur de pondération au modèle de langage, appelé *fudge factor*. La formule 1.3 devient :

$$\hat{W} \approx argmax_W P(W)^f \cdot P(X|W) \tag{1.17}$$

La valeur optimale de ce facteur est déterminée empiriquement sur un corpus de développement. La valeur choisie est celle qui optimise les performances du système de reconnaissance. En général, f>1.
Afin de déterminer la séquence W qui maximise l'équation 1.17, le système doit procéder à des multiplications successives des nombres compris entre 0 et 1. La capacité limitée d'un ordinateur de représenter un nombre proche de 0 est donc rapidement atteinte. C'est pourquoi, dans la pratique, les systèmes de reconnaissance de la parole ne manipulent pas directement les probabilités mais leurs logarithmes. Ainsi, le passage aux logarithmes entraîne l'utilisation des additions à la place des multiplications et permet de bénéficier de la propriété des logarithmes qui changent très lentement d'ordre de grandeur. L'équation 1.17 devient :

$$\hat{W} \approx argmax_W [f \cdot logP(W) + logP(X|W)] \tag{1.18}$$

1.5 Espace de recherche et sorties de reconnaissance

Un système de reconnaissance de la parole a pour but de générer, à partir d'un signal de parole et des connaissances *a priori* (lexique, modèle acoustique, modèle de langage,...), un ensemble d'hypothèses de séquences de mots. L'algorithme de décodage le plus souvent utilisé dans les systèmes de reconnaissance de la parole est basé sur le critère du maximum de vraisemblance (*Maximum A Posteriori*–MAP) qui détermine la séquence de mots qui maximise l'equation 1.3. Pour déterminer l'ensemble des hypothèses de reconnaissance, le décodeur explore un espace de recherche qui contient

les informations nécessaires à la génération des hypothèses : les unités acoustiques associées à leur scores acoustiques (donnés par le modèle acoustique), les informations temporelles, les informations liées au modèle de langage, etc..
Dans la majorité des systèmes, la taille de l'espace de recherche est très importante ce qui ralenti considérablement la recherche de la séquence de mots de probabilité maximale. Les hypothèses sont construites et évaluées au fur et à mesure du décodage ce qui permet de mettre en oeuvre une méthode efficace d'élagage (*pruning*). Celle-ci élimine les hypothèses les moins probables par rapport à la meilleure hypothèse à un instant donné. Ce type de recherche est aussi appelée recherche en faisceau (*beam search*) (Ney et Ortmanns, 2000; Ortmanns et Ney, 1997).
L'utilisation des modèles de langage de type *n-gramme* avec un N assez grand peut aussi ralentir considérablement la recherche de la séquence de mots de probabilité maximale. La solution la plus répandue, malgré sa simplicité de modélisation, est d'utiliser un modèle *n-gramme* plus réduit comme par exemple un bigramme (N=2) ou un trigramme (N=3).
Dans la pratique, l'utilisation des techniques comme le (*beam search*) et des bigrammes ou trigrammes ne garanti pas l'obtention de la solution la plus probable, mais dans la majorité des cas, la solution trouvée est un bon compromis entre la durée de traitement et la perte de précision. Un modèle de langage plus complexe peut être utilisé en deuxième passe sur l'ensemble des solutions trouvées. En effet, un système de reconnaissance peut produire une seule solution (la séquence de mots de probabilité maximale), appelée aussi *1-best*, mais aussi un ensemble de solutions possibles structurées sous la forme :

- d'une liste ordonnée de N meilleures solutions.
- d'une structure compacte appelée graphe de mots. Le but de cette structure est de fournir des alternatives pour les parties du signal acoustique pour lesquelles l'ambiguïté de la prononciation est forte (Ney et Ortmans, 1997).

1.5.1 Liste de N meilleures solutions

Appelée aussi liste N *best*, elle contient les N meilleures solutions du décodeur ordonnées en fonction de leur probabilité calculée au sens MAP. Comme la *1-best*, les solutions de la liste ne contiennent aucune information temporelle en ce qui concerne les instants de début ou de fin des mots de chaque séquence. On ne connaît pas non plus les vraisemblances acoustiques de chaque mot.
Sur une liste de N *best* on peut calculer des mesures de confiance (voir 2.3.2), faire du *rescoring* afin de réordonner les solutions de la liste. L'espace d'hypothèses que représentent les solutions de la liste peut être utilisé comme espace de recherche par des modules applicatifs en aval. On pourrait penser que plus N est grand, meilleures seront les performances de la liste de N *best*. Outre le fait que dans (Wessel et al., 2000) les auteurs ont montré que les performances de mesures de confiance sont dégradées à partir d'une certaine valeur de N ($N == 1000$), l'utilisation d'une liste avec un N trop grand n'est pas très efficace en termes de temps de calcul et des ressources mémoire nécessaires. Il convient donc de choisir une valeur de N suffisamment grande pour que

l'espace d'hypothèses soit assez riche, mais cette valeur ne doit pas générer un temps de calcul et des besoins de ressources trop importants.

1.5.2 Graphes de mots

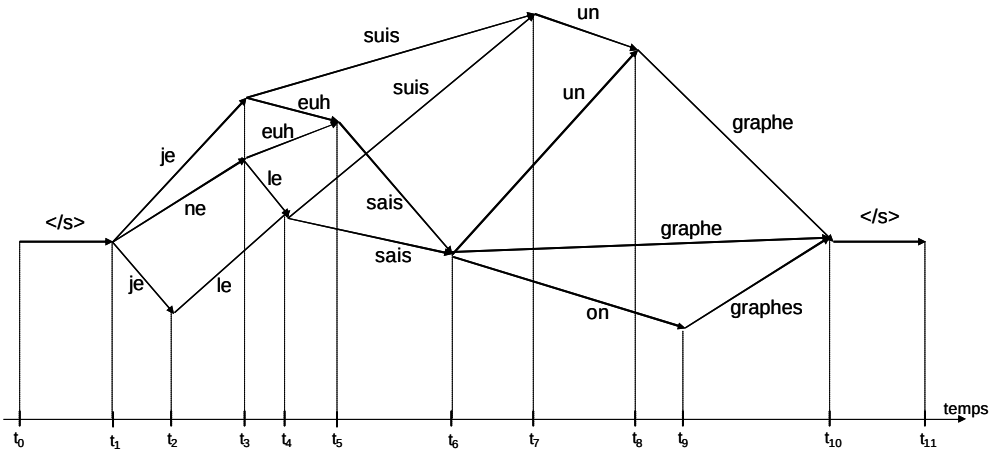

FIGURE *1.2* – Exemple simple de graphe de mots

La génération du graphe de mots se fait suite à un décodage de type Viterbi, synchrone dans le temps. A un instant donné, l'algorithme garde les hypothèses qui n'ont pas été éliminées par le *beam search*. La taille d'un graphe de mots dépend donc de la taille du faisceau, qui est un paramètre important dans la construction du graphe de mots.
La structure d'un graphe de mots est celle d'un graphe acyclique direct. La figure 1.2 montre un exemple simplifié. Deux éléments distincts composent le graphe de mots :

- les nœuds du graphe, aussi appelés états, sont caractérisés par l'instant temporel auquel ils ont été créés, appelé aussi trame de création d'un état.
- les arcs du graphe, aussi appelés transitions, sont caractérisés par un état de départ et un état de fin. Chaque transition porte une hypothèse de mot avec son score acoustique, qui mesure la vraisemblance entre les morceaux du signal acoustique correspondant à la transition et le modèle acoustique.

Les états du graphe sont numérotés en partant de 0 pour le premier état. Ainsi, du fait de sa structure acyclique, les états de fin des transitions se voit attribués un numéro plus grand que les états de début.

1.5.3 Réseaux de confusion

En 2000, une nouvelle structure, appelée réseau de confusion (CN - *Confusion Network*), a été proposée dans (Mangu et al., 2000) pour structurer l'ensemble des hypothèses en sortie de décodeur. Contrairement au graphe de mots, cette structure ne peut

pas être générée directement par le décodeur, mais seulement à partir d'un graphe de mots. Les algorithmes de génération sont détaillés au chapitre 5.

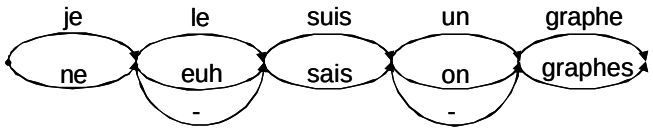

FIGURE *1.3* – Exemple de réseau de confusion. Le symbole "-" marque une omission

La figure 1.3 montre un exemple de réseau de confusion. On observe une structure similaire au graphe de mots par le fait que les éléments qui la constituent sont des transitions et des états. De plus, le CN garde la propriété d'être acyclique car on ne peut le parcourir que de gauche à droite sans qu'on puisse repasser deux fois par le même état. Comme pour les graphes de mots, les états sont numérotés et l'état de début d'une transition est plus petit que l'état de fin d'une transition. De plus, chaque état est caractérisé par un numéro de trame.
Comme le montre la figure 1.3, un réseau de confusion peut être assimilé a une séquence de "classes". Une classe est définie comme l'espace entre deux états consécutifs et contient une ou plusieurs transitions. Chaque transition est caractérisée par un état de début et un état de fin et se voit attachée une hypothèse de mot et sa probabilité *a posteriori*[2]. Une propriété importante d'une classe réside dans le fait que toutes les transitions de la classe partent du même état et finissent dans un même état (ce sont les deux états qui définissent la classe). Sur l'intervalle temporel représenté par la classe, les transitions sont donc en concurrence.
Du fait de sa structure, un chemin complet dans un CN (qui part de l'état 0 et arrive à l'état de fin) passe par tous les états du réseau et traverse ainsi toutes les classes en passant par une des transitions de chaque classe. En passant par une classe, le chemin peut soit passer par une transition portant un mot soit par une transition portant une "omission". Dans chaque classe on peut trouver au maximum une transition portant l'omission avec une probabilité *a posteriori* associée. Elle permet de traverser la classe sans passer par un mot. Les problématiques liées à cette transition, sa présence dans certaines classes et le calcul de la probabilité *a posteriori* associée sont discutées au chapitre 5.
Pour résumer, les éléments qui constituent un réseau de confusion sont :

- Les états caractérisés par un instant temporel. Les états sont numérotés en partant du premier état, à qui on attribue la valeur 0. Tout chemin complet du CN passe par tous les états en les traversant dans un ordre croissant de leur numéro.
- Les classes sont définies comme l'espace entre deux états consécutifs qui représentent les états de début et de fin de la classe. Tout chemin complet du CN traverse chaque classe une seule fois.
- Les transitions sont regroupées en classes, chacune pouvant en contenir une ou plusieurs transitions. Les états de début et de fin des transitions d'une classe sont identiques aux états de début et de fin de la classe. Chaque transition porte une hypothèse de mot avec la probabilité *a posteriori* associée. Un chemin complet du

2. Le calcul de cette probabilité est détaillé au chap 5

CN ne peut contenir qu'une seule transition de chaque classe. Dans une classe, il peut exister une transition portant une omission, avec une probabilité *a posteriori* associée, qui permet à un chemin complet de traverser une classe sans passer par un mot.

1.6 Évaluation des systèmes de reconnaissance de la parole

1.6.1 Taux d'erreur mot et *word accuracy*

Une fois la solution du système de reconnaissance obtenue, elle doit être évaluée afin de mesurer les performances du système. La mesure la plus souvent employée pour évaluer un système de reconnaissance de la parole et le taux d'erreur mot (*WER - Word Error Rate*). Ceci implique un comptage du nombre de mots mal reconnus par le système réalisé par une comparaison entre la meilleure solution du système (la séquence de mots reconnue) et la transcription manuelle du signal de parole (la séquence de mots réellement prononcée). La transcription manuelle du signal de parole est généralement appelée *transcription de référence* ou tout simplement *référence*.
Pour calculer le *WER*, les deux séquences de mots sont alignées et on mesure la distance de Levensthein, appelée aussi distance d'édition. Elle est égale au nombre minimal de mots qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre. Cette distance est d'autant plus grande que le nombre de différences entre les deux chaînes est grand.
Trois types d'erreurs possibles interviennent dans le calcul du *WER*.

- Les *omissions* sont les mots de la référence qui n'ont pas été reconnus par le système. Ils ne se retrouvent donc pas dans la solution fournie.
- Les *insertions* sont des mots reconnus par le système qui ont été insérés dans la solution en plus des mots de la référence.
- Les *substitutions* sont les mots qui ont été reconnus à la place d'autres mots de la référence.

Le *WER* est calculé en sommant les trois types d'erreurs et en normalisant par le nombre total de mots dans la référence :

$$WER = \frac{Nombre\ d'omissions\ +\ Nombre\ d'insertions\ +\ Nombre\ de\ substitutions}{Nombre\ de\ mots\ dans\ la\ référence} \tag{1.19}$$

Le *word accuracy* est défini comme étant égal à 1 moins la valeur du taux d'erreur mot.

1.6.2 Précision et rappel

D'autres métriques peuvent être utilisées pour mesurer les performances d'un système, comme par exemple la précision et le rappel, empruntés au domaine de la recherche d'informations.

La précision mesure la capacité du système de rejeter les hypothèses de mots incorrectes :

$$Précision = \frac{Nombre\ de\ mots\ correctement\ reconnus}{Nombre\ de\ mots\ reconnus} \tag{1.20}$$

Le rappel mesure la capacité du système à accepter les hypothèses de mots correctes :

$$Rappel = \frac{Nombre\ de\ mots\ correctement\ reconnus}{Nombre\ de\ mots\ dans\ la\ référence} \tag{1.21}$$

1.6.3 Taux d'erreur mot Oracle

Les métriques présentées se calculent seulement lorsque le résultat du système de reconnaissance est sous la forme d'une séquence d'hypothèses de mots. Mais certains systèmes sont aussi capables de produire en sortie des graphes de mots. Il est toujours possible d'extraire une séquence de mots du graphe (qui forme un chemin complet), généralement la meilleure solution, au sens de MAP, et de calculer le *WER*, la précision ou le rappel. Néanmoins, en raison des techniques utilisées lors de la construction du graphe, tel que le *beam search*, cette solution n'est pas forcément la meilleure solution que le système peut obtenir. Il est donc nécessaire d'utiliser une métrique qui permet de mesurer les performances du graphe de mots dans sa globalité, en prenant en compte l'ensemble des hypothèses présentes dans le graphe.
L'oracle est obtenu en recherchant dans le graphe de mots la séquence de mots la plus proche de la référence du point de vue de la distance d'édition. Le taux d'erreur mot Oracle (*WER* Oracle) est alors le meilleur taux d'erreur mot qu'on pourrait obtenir si on savait retrouver l'oracle de manière automatique. Il peut être vu comme la limite théorique du *WER* qui pourrait être obtenu en choisissant idéalement les séquences de mots dans le graphe. Ce taux peut servir à mesurer le potentiel d'un graphe de mots en termes de *WER*, mais aussi à comparer des graphes de mots obtenus avec différents systèmes de reconnaissance ou avec des paramètres différents d'un même système.
De la même manière que pour le *word accuracy* on peut définir un *oracle accuracy* qui est égal à 1 moins la valeur du taux d'erreur oracle.

1.7 Conclusions

Dans ce chapitre nous avons tout d'abord présenté les différents modules qui composent un système de reconnaissance de la parole et leur fonctionnement. Nous avons ainsi décrit le module d'analyse acoustique du signal qui a pour but de transformer le signal acoustique en une suite de vecteurs de coefficients utilisés ensuite dans le processus de décodage. Ce processus est réalisé par le décodeur de parole qui utilise des modèles acoustiques et de langage que nous avons également décrits. La problématique liée à la combinaison des deux modèles a aussi été discutée. Nous avons ensuite abordé les problématiques liées à l'espace de recherche utilisé par le décodeur. Une description de l'espace d'hypothèses en sortie de décodeur et les différentes structures utilisées ont

été abordées par la suite. La dernière partie du chapitre a présenté les différentes métriques utilisées pour l'évaluation des performances d'un système de reconnaissance de la parole.