ANTS le système de transcription automatique du LORIA

Armelle Brun, Christophe Cerisara, Dominique Fohr, Irina Illina, David Langlois, Odile Mella

Equipe Parole, LORIA, Vandoeuvre-lès-Nancy

{brun,cerisara,fohr,illina,langlois,mella}@loria.fr
http://www.loria.fr/equipes/parole

Abstract

In the context of the ESTER project, we have developed the ANTS system (Automatic News Transcription System). This paper presents the different modules: telephone-speech/broadband-speech segmentation, speech/non-speech segmentation, breath detection, speaker clustering, recognition engine.

1. Introduction

Le projet Technolangue EVALDA-ESTER a pour objet l'évaluation des systèmes de transcription automatique d'émissions radiophoniques francophones. Dans ce cadre, l'équipe Parole a développé le système ANTS (Automatic News Transcription System).

Dans cet article, nous allons d'abord présenter brièvement l'architecture générale de notre système. Puis nous décrirons le prototype réalisé spécifiquement pour la campagne EVALDA-ESTER, avant de détailler quelques expérimentations particulières.

2. Le système ANTS

Comme le montre la Figure 1, le système de transcription ANTS est fondé sur la combinaison d'un certain nombre de modules. Nous allons donner dans les paragraphes suivant les grands principes de ces modules.

2.1. La segmentation téléphone/non téléphone

Cette segmentation est réalisée à l'aide de deux mélanges de lois gaussiennes (GMM). Le premier modèle est appris sur de la parole « bande étroite » c'est-à-dire prononcée au téléphone. L'autre GMM est appris sur le reste des données disponibles.

La segmentation est fondée sur l'algorithme de Viterbi. Afin d'éviter l'alternance de segments différents de trop courte durée, non représentative de la réalité, la reconnaissance n'utilise pas directement les deux GMS appris mais deux modèles de Markov, chacun étant composé de N états identiques. Chaque segment obtenu, téléphonique ou non téléphonique » aura donc une longueur minimale de N trames.

2.2. La segmentation parole/non-parole

La partie non téléphonique du signal audio subit ensuite une segmentation parole/non-parole fondée sur la mise en compétition de quatre modèles GMMs : Parole, Musique Instrumentale, Chant et enfin Parole-Musique, modélisant la superposition de la parole et de la musique.

Comme dans le module précédent, la phase de reconnaissance utilise l'algorithme de Viterbi et impose une durée minimale pour chaque segment grâce à la concaténation de N modèles GMMs identiques.

Les segments reconnus comme Musique Instrumentale ou Chant sont éliminés définitivement, seule sera transcrite la parole seule ou la parole sur fond musical. Notons également que nous avons fait l'hypothèse que la partie « bande étroite » du signal audio ne comportait pas des morceaux de « pure musique » ou de « pures chansons ».

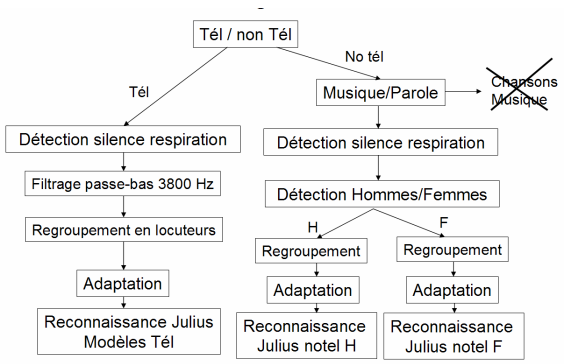

Figure 1 : Architecture générale du système ANTS.

2.3. Détection des respirations et des silences

Quel que soit le type de parole, téléphonique ou non, nous avons choisi à cette étape de détecter les pauses et les respirations effectuées par les locuteurs. Ceci en vue de deux objectifs :

- découper la parole en morceaux de plus petite taille pour le moteur de reconnaissance,
- trouver les groupes de souffle qui correspondent souvent à des entités syntaxiques ou sémantiques.

Pour réaliser une telle segmentation, nous procédons à une reconnaissance au niveau phonétique en utilisant des modèles de phonèmes, un modèle de silence et un modèle de respiration ou de souffle. La grammaire utilisée pendant cette reconnaissance attribue la même probabilité à toutes les transitions entre les modèles.

Seuls les respirations et les silences de taille suffisante sont pris en compte et toute portion de signal comprise entre deux respirations et/ou silences est alors extraite.

2.4. Segmentation hommes/femmes

Ce module est similaire au module de segmentation "téléphone/non téléphone" (cf § 2.1) : un modèle GMM est appris avec de la parole prononcée par des locuteurs masculins, un autre par des locuteurs féminins.

De la même manière que dans les modules précédents, une concaténation des modèles permet d'éviter une alternance homme/femme irréaliste.

2.5. Regroupement par locuteurs

Ce module regroupe les segments de parole prononcés par des locuteurs ayant des voix similaires afin d'effectuer une adaptation dans une étape ultérieure.

L'algorithme regroupant deux segments est fondé sur le critère BIC (*Bayesian Information Criterium*) et s'inspire d'une méthode proposée par Perrine Delacourt [8] :

- chaque segment est représenté par une loi gaussienne,
- pour chaque couple de segments, le BIC est calculé,
- le couple de segments qui maximise le BIC est fusionné en un segment unique.

Le processus est itéré tant qu'il existe des valeurs de BIC positives. Chaque segment ainsi obtenu est considéré comme ayant été prononcé par un seul locuteur (ou par des locuteurs ayant des voix proches).

2.6. Adaptation des modèles phonétiques

L'objectif de ce module est d'adapter au locuteur, de façon non supervisée, les modèles phonétiques qui sont utilisés par le moteur de reconnaissance. Pour cela, on effectue d'abord une première reconnaissance des segments obtenus à l'étape précédente (regroupement en locuteurs) en utilisant les modèles phonétiques non adaptés. Grâce à cette reconnaissance, ces modèles sont ensuite adaptés à l'aide d'une méthode de type MLLR ou MAP.

2.7. Moteur de reconnaissance

La phase de transcription proprement dite est réalisée à partir du moteur de reconnaissance Julius développé par Akinobu Lee [1]. Ce logiciel effectue la reconnaissance en deux passes : la première passe est trame-synchrone, utilise un modèle de langage bigramme et fournit un treillis de mots. La deuxième passe est fondée sur un algorithme à pile (de type A*) et utilise un modèle de langage trigramme. Le système Julius offre la possibilité de définir plusieurs prononciations pour chaque mot du lexique.

3. Mise en œuvre de la segmentation pour la campagne Ester

3.1. Préliminaires

Dans le cadre de la campagne d'évaluation Ester, des données radiophoniques transcrites ont été distribuées en deux étapes. Elles proviennent d'émissions d'informations des radios France-Inter, France-Info, RFI et RTM. Dans le suite de l'article, nous appellerons

- EsterTrainDry les 40 heures d'émissions distribuées au début de la campagne et qui ne concernent que France-Inter et RFI,
- EsterTrainTotal les 90 heures d'émissions comprenant EsterTrainDry complétées par 50 heures distribuées ultérieurement.

Par ailleurs, la paramétrisation de base effectuée sur le signal audio par tous les modules est un calcul des 13 coefficients MFCC avec ou sans MCR (Mean Cesptrum Removal), avec ou sans ajout des dérivées premières (Δ) et secondes ($\Delta\Delta$). Le calcul a été sur des fenêtres de 32 ms décalées de 10 ms. 24 filtres répartis sur une échelle Mel sont utilisés. Si une variante a été appliquée dans un module elle sera spécifiée dans le paragraphe correspondant.

3.2. La segmentation téléphone/non téléphone

Le signal audio est paramétrisé par 13 coefficients MFCC sans MCR afin de conserver l'influence du canal de transmission. Le vecteur d'observation ne comporte donc que 13 coefficients.

L'apprentissage des deux GMMs « (téléphonique/non-telephonqiue) a été effectué grâce au logiciel HTK sur 31 heures du corpus EsterTrainDry qui avait été étiquetées manuellement en « bande étroite » et « bande large » par le laboratoire CLIPS. Chaque GMM a 32 gaussiennes.

Afin d'assurer une durée minimale d'une demi-seconde pour chaque segment détecté, chaque modèle utilisé lors de la reconnaissance est la concaténation de 50 GMMs.

3.3. La segmentation parole/ non-parole

Le signal audio est paramétrisé par 13 coefficients MFCC + Δ + $\Delta\Delta$. Les quatre modèles GMM en compétition ont été appris :

- Sur 8H30 de segments du corpus EsterTrainDry étiquetés F0 (8 heures 30) pour le modèle « Parole »,
- Sur 1H45 de segments du corpus EsterTrainTotal étiquetés F3 pour le modèle « parole sur fond musical »,
- Sur 48 minutes des pistes audio extraites de CD musicaux pour le modèle « musique instrumentale »,
- Sur 1H30 de pistes audio extraites de CD de chansons pour le modèle « chant ».

Afin d'assurer une durée minimale d'une demi-seconde, 50 GMMs à 16 gaussiennes ont été concaténés pour la reconnaissance.

3.4. Détection des respirations et des silences

Le signal audio est paramétrisé par 13 coefficients MFCC + Δ + $\Delta\Delta$. Nous avons utilisé MCR pour réduire l'influence du microphone et du canal de transmission.
Ce module étant utilisé pour les deux types de parole téléphonique et non téléphonique :

- 36 modèles de phonèmes, un modèle de silence et un modèle de respiration ont été appris sur 15 heures « bande large » du corpus EsterTrainDry.
- 36 modèles de phonèmes, un modèle de silence et un modèle de respiration ont été appris sur 3 heures « bande étroite » du corpus EsterTrainDry.

Tous ces modèles ont 3 états, sans saut et chaque état comporte 2 gaussiennes.

Les segments sont découpés dans le cas où ils sont séparés par une respiration qui dure au minimum 150 ms ou par un silence d'au moins 300 ms.

3.5. Segmentation hommes/femmes

Le signal audio est paramétrisé par 13 coefficients MFCC + Δ + $\Delta\Delta$ avec MCR. Le modèle GMM « hommes » (respectivement « femmes ») a été appris avec les segments étiquetés « M » (respectivement « F ») de la transcription des

fichiers de EsterTrainDry, ce qui représente 33 heures (respectivement 16 heures) de parole. Chaque GMM est un mélange de 256 gaussiennes.

Lors de la reconnaissance, afin d'assurer une durée minimale d'une demi-seconde par segment, chaque modèle est la concaténation de 50 GMMs.

3.6. Regroupement par locuteurs

Le signal audio est paramétrisé par 13 coefficients MFCC + $\Delta + \Delta\Delta$ sans MCR afin de ne pas atténuer l'influence du canal de transmission et l'environnement acoustique propre à chaque locuteur.

Pour cette première version minimaliste du module de regroupement, nous avons choisi de modéliser chaque *cluster* par une gaussienne dont la matrice de covariance est supposée diagonale.

3.7. Adaptation

Pour chaque cluster, qui est censé ne contenir la parole que d'un seul locuteur ou de locuteurs ayant des voix similaires, on effectue une adaptation SMLLR « block-diagonal » des modèles acoustiques. Nous utilisons 3 blocs pour la matrice de transformation qui correspondent aux coefficients statiques, aux dérivées premières et aux dérivées secondes.

Cette adaptation est non supervisée et la suite de phonèmes prononcés est obtenue par une reconnaissance « grand vocabulaire » à l'aide du moteur Julius et des modèles acoustiques non adaptés. Les caractéristiques du moteur et les modèles utilisés sont décrits dans le paragraphe suivant.

3.8. Moteur de reconnaissance

Nous avons utilisé le moteur de reconnaissance Julius développé par Akinobu Lee. Pour la tâche « temps-réel » de la campagne Ester, nous avons utilisé directement la version « Fast » alors que pour la tâche « transcription » nous avons fait une modification de la version « Standard » pour améliorer la deuxième passe de la recherche. Nous avons modifié le critère du tri des hypothèses pour l'algorithme A* : les hypothèses sont triées d'abord sur le temps de début du premier mot de l'hypothèse, puis sur le score de l'hypothèse. De ce fait, l'algorithme explore les hypothèses d'une manière qui est proche de celle de l'algorithme « largeur d'abord ».
Les modèles acoustiques utilisés par le moteur sont décrits dans la section suivante.

4. Modèles acoustiques pour la transcription dans le cadre de la campagne Ester

Le signal audio est paramétrisé par 13 coefficients MFCC + $\Delta + \Delta\Delta$ avec MCR. De plus, le coefficient C0 a été remplacé par le logarithme de l'énergie.
Pour réaliser l'apprentissage des modèles acoustiques, le corpus EsterTrainTotal a été étiqueté phonétiquement de la manière suivante :

- Génération des prononciations possibles des phrases du corpus. Pour cela, plusieurs prononciations possibles ont été fournies par le dictionnaire phonétique BDLEX [9] ou des dictionnaires ad-hoc élaborés par l'équipe, une pause optionnelle entre les mots a été ajoutée et toutes les liaisons ont été considérées comme facultatives.
- Alignement des prononciations possibles avec le signal audio et des modèles HMM phonétiques, la prononciation qui obtient la vraisemblance maximale nous permet d'obtenir la séquence de phonèmes prononcés.

Grâce à ce corpus, nous avons appris des modèles HMM à l'aide du logiciel HTK:

- 36 modèles phonétiques à trois états,
- 1 modèle correspondant à une courte pause,
- 1 modèle de respiration,
- 1 modèle de « bruit de bouche »,
- 1 modèle correspondant à tous les autres bruits.

Tous ces modèles ont trois états, sauf le modèle de pause courte qui n'en comporte qu'un seul.

De façon classique, à partir de ces modèles « hors contextes » à une gaussienne, un arbre de décision est créé afin d'obtenir des modèles triphones.

4.1. Modèles bande large

Pour extraire les phrases correspondant à « la bande large » du corpus EsterTrainTotal, nous avons utilisé notre système automatique de segmentation « tel/non tel » fondé sur les GMM (cf § 2.1). Nous n'avons pas utilisé l'étiquetage manuel (F0, F1, F3…) fourni avec le corpus car les transcripteurs n'avaient pas accès au spectrogramme pour décider « bande large / bande étroite », mais pouvaient seulement écouter le signal audio.

Au final 89490 gaussiennes sont estimées sur 62 heures de EsterTrainTotal « bande large ». Puis, les modèles dépendant du genre sont obtenus en utilisant 8 itérations d'une adaptation SMAP.

4.2. Modèles bande étroite

Afin d'augmenter artificiellement la quantité de données audio « bande étroite », nous avons filtré (entre 150-3800Hz) l'intégralité du corpus EsterTrainTotal. Les modèles triphones ont été appris sur ce corpus filtré et au total 115200 gaussiennes ont été estimées. Puis, une adaptation MAP a été effectuée sur la partie « bande étroite » de EsterTrainTotal.

5. Lexique et modèle de langage pour la campagne Ester

Le lexique est constitué des 5000 mots les plus fréquents du corpus EsterTrainTotal+EsterDev et des 55000 mots les plus fréquents du corpus textuel du journal « Le Monde » 1987-2003. Les noms communs ont été phonétisés grâce à BDLEX et les phonétisations des noms propres ont été réalisées par un phonétiseur automatique puis corrigées manuellement. Au final, le lexique contient 112000 prononciations.

Le modèle de langage est obtenu par interpolation linéaire des deux modèles suivants :

- Modèle trigramme généré à partir du corpus textuel « Le Monde », la méthode de repli est *witten-bel*l et un valeur du *cut-off* est 1.
- Modèle trigramme appris à l'aide du corpus EsterTrainTotal+EsterDev sans *cut-off*. La méthode de repli choisie est *absolute*.

Les poids pour l'interpolation sont 0.55 pour le modèle de langage « Le Monde » et 0.45 pour l'autre modèle de langage. Au final, le modèle de langage contient 7.4 millions de bigrammes et 25.4 millions de trigrammes.

6. Expérimentations sur le lexique et le modèle de langage

Nous avons réalisé des expérimentations pour étudier l'influence du modèle de langage sur la perplexité. Les modèles de langage sont construits à partir de la combinaison linéaire entre trois modèles, chacun associé à un corpus d'apprentissage distinct :

- Corpus journalistique : il s'agit des données du journal Le Monde fournies lors de la campagne,
- Corpus RadioVariée: il s'agit de données radiophoniques diverses,
- Corpus radiophonique : il s'agit d'émissions d'informations de France Inter, Radio France, RFI et RTM fournies lors de la campagne ESTER.

Tous les modèles utilisent le même vocabulaire dont le contenu a été construit en utilisant les données des trois sources. Pour chaque modèle, nous avons déterminé un ensemble de paramètres optimaux (méthodes de repli, suppression des événements rares…) qui peuvent différer de l'un à l'autre.

Nous avons généré le modèle de langage pour l'évaluation finale en deux temps :

1. Utilisation des corpus fournis pour la première session d'évaluation (test à blanc), séparés en un corpus d'apprentissage, un corpus de développement et un corpus de test pour :

- Estimer les poids relatifs de chaque source de corpus (corpus journalistique, RadioVariée, données radiophoniques) dans la création du vocabulaire ;
- Détermination des méthodes de repli (*discounting*) et des paramètres de *cut-offs* optimaux pour chaque modèle de langage [2] ;
- Estimation des paramètres optimaux pour la combinaison linéaire entre les trois modèles de langage.

2. Réutilisation de ces paramètres avec les corpus fournis pour la campagne finale (plus corpus RadioVariée), en utilisant cette fois-ci tous les corpus en tant que corpus d'apprentissage.

Ce sont ces deux phases que nous décrivons successivement ci-dessous.

6.1. Phase 1 : détermination des paramètres optimaux

L'objectif est ici d'utiliser les méthodes classiques en modélisation statistique du langage pour estimer les paramètres de notre modèle de langage.

Pour obtenir les meilleurs modèles possibles, nous pouvons agir sur : le vocabulaire, les méthodes de repli et les *cut-offs*. Nous ne pouvions pas tester toutes les possibilités pour toutes ces dimensions. Nous avons donc dans un premier temps fixé le vocabulaire et cherché les meilleures stratégies de repli et valeurs de *cut-off* pour chaque modèle. Puis, nous avons utilisé ces valeurs optimales pour tester plusieurs vocabulaires.

6.1.1. Génération des corpus

Pour générer les corpus d'apprentissage, de développement et de test, nous avons utilisé les trois sources de données textuelles décrites ci-dessus. Au moment de ces tests, nous avions à disposition des volumes de données décrits dans Tableau 1.

Tableau 1:Volume des données source.

Nom du corpus	Taille (en mots)
Le Monde	360M
RadioVariée	34M
France Inter	300K
France Info	300K
RFI	300K

Nous avons utilisé tout le corpus « Le Monde » pour le corpus d'apprentissage `train_le_monde`. Le corpus « RadioVariée » a été entièrement utilisé pour le corpus d'apprentissage `train_radiovariée`. Les trois corpus de radio France Inter, France Info et RFI ont été utilisés pour générer un corpus d'apprentissage `train_radio`, un corpus de développement `dev_radio` et un corpus de test, `test_radio`. Nous avons fait ce choix afin que les données de développement et de test soient les plus proches possibles des données de test de l'évaluation finale. Chaque fichier de radio a participé pour un tiers à chacun des trois corpus `train_radio`, `dev_radio` et `test_radio`. Finalement les tailles de chacun de ces corpus, supports des modèles de langage, sont données dans Tableau 2.

Tableau 2 : Volumes des corpus de la phase 1

Statut	Nom du corpus	Taille
Apprentissage	train_le_monde	360M
	train_radiovariée	34M
	train_radio	300K
Développement	dev_radio	300K
Test	test_radio	300K

6.1.2. Recherche des meilleures techniques de repli, choix des cut-offs

Pour opérer une combinaison linéaire des trois modèles trigrammes issus des trois corpus d'apprentissage, nous avons cherché à utiliser les modèles trigrammes les plus performants en terme de perplexité. Pour cela, nous avons cherché la meilleure technique de repli et choix de *cut-off* pour chaque modèle.

Nous avons testé quatre méthodes de repli : *linear*, *absolute*, *good-turing* [2] et *witten-bell* [7] et 10 valeurs de *cut-off* (de 0 à 10). Pour chaque corpus d'apprentissage, chaque méthode de repli et chaque valeur de *cut-off*, nous avons généré un modèle trigrammes. Pour cette recherche, nous avons utilisé systématiquement le même vocabulaire contenant les 60000 mots les plus fréquents de `train_le_monde`. Nous avons testé tous les modèles sur le corpus de développement (`dev_radio`).

Nous avons établi que :

- Pour `train_le_monde` et `train_radiovariée`, la meilleure technique de repli est *witten-bell* (bien que la meilleure méthode ne soit pas systématiquement significativement la meilleure). Les Tableaux 3 et 4 fournissent pour chaque méthode de repli, le meilleur résultat en terme de perplexité (sur dev_radio);

- Pour `train_radio`, la meilleure technique de repli est *absolute*. Voir Tableau 5 (les remarques données pour les deux tableaux précédents restent valables).

- Pour ces méthodes, mieux vaut ne pas appliquer de *cut-off*. Un *cut-off* de 1 a toutefois été imposé pour `train_le_monde` du fait de la grande taille du corpus d'apprentissage. La figure 2 montre l'évolution de la perplexité sur le corpus de développement en fonction de la valeur du *cut-off*.

Tableau 3 : Performances des méthodes de repli pour train_le_monde.

Méthode	PP	Cut-off
Linear	138.5	1
Absolute	132.5	1
Good turing	133.8	1
Witten-bell	**132.0**	**1**

Tableau 4 : Performances des méthodes de repli pour train_radiovariée

Méthode	PP	Cut-off
Linear	224.6	1
Absolute	214.7	0
Good turing	219.5	0
Witten-bell	**208.4**	**0**

Tableau 5: Performances des méthodes de repli pour train_radio

Méthode	PP	Cut-off
Linear	325.0	0
Absolute	**273.3**	**0**
Good turing	278.9	0
Witten-bell	284.6	0

6.1.3. Recherche des paramètres du vocabulaire

Nous avons choisi arbitrairement 60K mots pour la taille du vocabulaire. Pour choisir le vocabulaire, nous avons testé deux méthodes.

La première a consisté à choisir les mots du vocabulaire parmi les mots de `train_le_monde`, `train_radiovariée` et `train_radio` selon un critère de maximum de vraisemblance sur le corpus de développement `dev_radio`. Pour ce faire, nous avons utilisé l'outil `select-vocab` du SRI et choisi les 60000 mots possédant le meilleur score selon cet outil. Nous appelons cette méthode VOCVRAIS.

La deuxième méthode consiste à sélectionner les mots les plus fréquents de chacun des trois corpus `train_le_monde`, `train_radiovariée`, `train_radio`. Pour cela, il faut ordonner les trois corpus en `train_1`, `train_2`, `train_3`, choisir deux valeurs `taille_1` et `taille_2` (`taille_1+taille_2<60000`) et suivre l'algorithme ci-dessous :

- Soit `v` une liste vide
- insérer dans `v` les `taille_1` mots les plus fréquents de `train_1`
- construire `v_2` la liste des mots de `train_2` n'apparaissant pas dans `v`
- si la taille de `v_2 >= taille_2`, insérer dans `v` les `taille_2` mots les plus fréquents de `v_2` sinon sortir en erreur pour choisir d'autres paramètres `taille_1` et `taille_2`
- construire `v_3` la liste des mots de `train_3` n'apparaissant pas dans `v`
- si la taille de `v_3 >= 60000-taille_2-taille_3`, insérer dans `v` les `taille_3` mots les plus fréquents de `v_3` sinon sortir en erreur pour choisir d'autres paramètres `taille_1` et `taille_2`
- `v` est le résultat

Nous appelons cette méthode VOCFREQ. Cette méthode est paramétrée par la fonction de correspondance entre l'ensemble `{train_le_monde, train_radiovariée, train_radio}` et l'ensemble `{train_1, train_2, train_3}`, ainsi que par `taille_1` et `taille_2`. Nous avons testé plusieurs fonctions de correspondance et opéré une recherche de type « grid search » pour `taille_1` et `taille_2`.

Pour chaque vocabulaire, nous avons évalué un modèle trigrammes composé de la combinaison linéaire entre les trois modèles trigrammes correspondant respectivement à `train_le_monde`, `train_radiovariée` et `train_radio`. Pour ces modèles, nous avons utilisé les paramètres optimaux déterminés dans la section précédente. Les poids de la combinaison linéaire ont été estimés par maximum de vraisemblance sur le corpus de développement `dev_radio`.

Le vocabulaire obtenu grâce à la méthode VOCVRAIS obtient une perplexité égale à 92.8 sur le corpus de test `test_radio`.

Les vocabulaires issus de la méthode VOCFREQ obtiennent des perplexités qui évoluent entre 85.9 et 91.0. Le vocabulaire obtenant la perplexité la plus petite (85.9) est composé de 60000 mots provenant exclusivement de `train_radiovariée` (ou 59000 venant de `train_radiovariée` et 1000 mots provenant de `train_le_monde`). Pour ces deux derniers vocabulaires, les poids de la combinaison linéaires des modèles de langage sont respectivement 0.56, 0.20, 0.24 pour `train_le_monde`, `train_radiovariée` et `train_radio`.

6.2. Travaux sur les séquences

Nos précédents travaux [4,6] ont montré l'apport de l'utilisation de séquences dans les systèmes de reconnaissance. Ces travaux sont en accord avec ceux provenant d'autres chercheurs [3,5]. Nous avons donc décidé

d'inclure dans le vocabulaire des suites de mots composées des mots du vocabulaire initial. Nous avons utilisé la méthode de l'information mutuelle [7] pour déterminer un ensemble de séquences. Nous ne donnerons pas plus de précisions sur le protocole d'intégration et de test car, étonnement, les séquences n'ont pas donné les résultats escomptés sur le WER.

En revanche, nous avons voulu reprendre le problème à la source en faisant une étude de cas sur un petit ensemble de séquences qui nous paraissaient intéressantes. Nous avons travaillé sur le fichier 7H00-8H00 France Inter du 18 avril 2003 et choisi un ensemble de séquences fréquentes, mais mal reconnues. Notre choix s'est porté sur les suites de mots ayant pour pivot le mot « y » (voir Tableau 6 qui montre que la séquence « il y a » est présente 27 fois dans le fichier de référence, et qu'elle a été reconnue correctement seulement dans 25.9% des cas).

Puis nous avons créé un nouveau vocabulaire et modèle de langage (mêmes paramètres que ceux déterminés dans les sections précédentes) intégrant les séquences de 3 mots les plus fréquentes (dans le corpus d'apprentissage) ayant pour mot central « y ». Nous avons relancé le système avec ces nouvelles données. Nous donnons dans Tableau 7 les nouvelles performances obtenues. On remarque tout de suite que la séquence « il y a » est bien mieux reconnue (66.7%), mais que la reconnaissance de « il y avait » chute. Le fait que « avait » commence par « a » n'est sans doute pas étranger à cette chute.

Globalement, l'amélioration en performances n'est pas significative (gain absolu de 0.1% pour le WER sur l'ensemble du fichier de référence). Face à cette étude de cas, il faut rester prudent. En effet, ce travail peu coûteux en temps a permis d'améliorer le taux de reconnaissance de la suite très fréquente « il y a » ; travailler sur quelques séquences peut donc avoir un impact. Mais déjà, cela aboutit par ailleurs à d'autres erreurs (sur « il y avait »). Il semble difficile de continuer sur cette voie, en traitant ainsi le problème au cas par cas, puisqu'un phénomène est rarement indépendant des autres. Par ailleurs, pour cette étude, nous avons choisi les séquences qui nous semblaient les plus prometteuses. Les étapes suivantes concerneraient donc des suites moins intéressantes, *a priori*.

En conclusion, il faudra déterminer pourquoi les méthodes classiques d'introduction massive de séquences n'aboutissent pas aux mêmes améliorations sur cette application (cela vient-il de la nature de l'application, du système, de la redondance avec les modèles acoustiques ?...). Par ailleurs, il semble que l'introduction de séquences au cas par cas peut être utile. Mais cela demandera une gestion non triviale des poids de ces nouvelles séquences dans le processus de reconnaissance, afin que les apports ne soient pas contrebalancés par un accroissement en confusion.

Tableau 6 : Performances en reconnaissance de suites de mots ayant pour pivot le mot "y" (système n'utilisant pas de séquences)

Suites de référence	**Nombre d'occurrences dans la référence**	**Distribution en pourcentage des hypothèses suite à la reconnaissance ([] correspond à une élision totale de la suite de référence)**
il y a	27	[il y a] 25.9 [] 14.8 [il] 7.4 [autres] 3.7 [bien] 3.7 [comme ça a] 3.7 [dites rien] 3.7 [elle] 3.7 [filiale de] 3.7 [hier] 3.7 [il y] 3.7 [lire] 3.7 [problème c' est] 3.7 [que les à] 3.7 [ses m' a] 3.7 [y a] 3.7 [à] 3.7
n' y a	5	[n' y a] 40.0 [finir] 20.0 [que] 20.0 [y a] 20.0
il y avait	4	[il y avait] 50.0 [] 25.0 [bien mais] 25.0
il y aura	3	[] 33.3 [exclure] 33.3 [leur a] 33.3
n' y en	2	[va y en] 50.0 [y] 50.0
il y en	1	[] 100.0
doit y avoir	1	[] 100.0

Tableau 7 : Performances en reconnaissance de suites de mots ayant pour pivot le mot "y" (système utilisant des séquences)

Suites de référence	**Nombre d'occurrences dans la référence**	**Distribution en pourcentage des hypothèses suite à la reconnaissance ([] correspond à une élision totale de la suite de référence)**
il y a	27	[il y a] 66.7 [] 11.1 [autres] 3.7 [de ce que] 3.7 [elle] 3.7 [problème c' est] 3.7 [ses m' a] 3.7 [à] 3.7
n' y a	5	[n' y a] 40.0 [y a] 40.0 [finir] 20.0
il y avait	4	[] 25.0 [il y] 25.0 [il y a les] 25.0 [il y avait] 25.0
il y aura	3	[il y aura] 66.7 [exclure] 33.3
n' y en	2	[y] 50.0 [y en] 50.0
il y en	1	[] 100.0
doit y avoir	1	[] 100.0

7. Résultats expérimentaux

Nous avons extrait les 10 premières minutes d'un fichier de France-Info pour réaliser des tests comparatifs

7.1. Influence du nombre de gaussiennes

Le tableau 8 montre l'influence du nombre de gaussiennes des modèles triphones sur le taux d'erreur en mot.

Tableau 8 : Influence du nombre de gaussiennes

Nombre de gaussiennes	Taux d'erreur WER %
17898	24.2
29830	24.3
41762	22.4
59660	21.0
89490	21.1
119276	21.7

7.2. Influence de l'algorithme de recherche en 2e passe

Le taux d'erreur en mot (WER) en utilisant l'algorithme A* par défaut de Julius est de 20.4%. Après avoir modifié l'algorithme en le rapprochant d'un algorithme en largeur d'abord, nous obtenons un taux de 18.4%.

7.3. Influence du poids du modèle de langage

Lors de la reconnaissance, Julius calcule la vraisemblance d'une trame ou d'une phrase en combinant le score acoustique (SA) et le score linguistique (SL) de la façon suivante :

$$Log(V)= SA+ (SL*K_{p1}+K_{p2})$$

Durant la première passe les poids K_{11} et K_{12} sont utilisés ; de la même façon K_{21} et K_{22} sont utilisés pendant la deuxième passe. Le tableau 9 montre l'influence non négligeable du réglage de ces poids.

Tableau 9: Influence des poids entre les deux scores

1ère passe		2ème passe		Taux d'erreur WER %
K_{11}	K_{12}	K_{21}	K_{22}	
8	-8	8	-8	19.5
12	-8	14	-12	26.6

Les performances de reconnaissance sont donc très dépendantes de la pondération entre modèle acoustique et modèle de langage. Le réglage de cette pondération est longue et fastidieuse.

8. Le système Temps-réel

8.1. Description du système

Le système temps-réel diffère du système précédemment décrit dans l'article par les éléments suivants:

- Julius est compilé en mode "Fast" et l'algorithme de recherche de deuxième passe n'a pas subi de modification.
- le faisceau de recherche lors de la première passe est plus fin,
- il n'y a ni regroupement ni adaptation au locuteur,
- seulement 41762 gaussiennes au total pour les modèles « bande large »,
- seulement 23040 gaussiennes au total pour les modèles « bande étroite ».

8.2. Résultats expérimentaux

Nous avons essayé deux approches différentes : soit des modèles acoustiques plus précis (nombre de gaussiennes) mais avec un faisceau de recherche plus fin, soit le contraire.

Le résultat de cette étude est donné dans le tableau 10.

Tableau 10: Influence du nombre de gaussiennes et du faisceau de recherche.

Taille du faisceau	Nombre de gaussiennes	Taux d'erreur WER %
600	47728	21.8
800	35796	20.4
1000	29830	20.7

9. Conclusion

Grâce aux ressources distribuées lors de la campagne Ester nous avons pu construire un système complet pour la transcription d'émissions radiophoniques. Nous envisageons de nombreuses améliorations : affiner la segmentation parole/musique, améliorer la segmentation et le regroupement par locuteur, prendre en compte les données non transcrites pour l'apprentissage des modèles acoustiques, intégrer des quadrigrammes dans Julius.

10. Remerciements

Ce travail a pu être mené grâce à l'action ministérielle Technolangue EVALDA-ESTER.
Nous remercions les équipes GEOD du CLIPS et TALNO du LIA pour la fourniture d'étiquetages manuels.

11. References

[1] A. Lee, T. Kawahara, and K. Shikano, "Julius – on open source real-time large vocabulary recognition engine", Eurospeech, p. 1691-1694.

[2] Federico M. and De Mori R. Spoken dialogues and Computers, chapter Language Modelling. p. 199-220, Academic Press, 1997.

[3] Kuo H.-K. and Reichl W. *Phrase-based language models for speech recognition*. In Proceedings of the European Conference on Speech Communication and Technology, p. 1595-1598, 1999.

[4] Langlois D., Smaïli K. and Haton J.-P. *Retrieving phrases by selecting the history: application to Automatic*

Speech Recognition. In 7th International Conference on Spoken Language Processing. 2002.

[5] Suhm S. and Waibel A. *Towards better language models for spontaneous speech*. In Proceeding of International Conference on Spoken Language Processing, p. 831-834, 1994.

[6] Zitouni I., Smaïli K. and Haton J.-P. Statistical language modelling based on variable-length sequences. Computer Speech and Language, vol 17, n°1, p. 27-41, 2003.

[7] Witten I. H. and Bell T. C. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in AdaptiveText Compression. in IEEE Transactions on Information Theory, vol 37, No. 4, July 1991.

[8] Delacourt, P. "la segmentation et le regroupement par locuteurs pour l'indexation de documents audio" Thèse de l'Ecole Nationale Supérieure des Télécommunications, 2000.

[9] M. De Calmès, G. Pérennou, « BDLEX : a Lexicon for Spoken and Written French », LREC 98, Grenade, p. 1129-1136, 1998.

Figure 2 : Performances de la meilleure méthode de repli pour chaque corpus d'apprentissage en fonction de la valeur de cut-off