Sirocco, un système ouvert de reconnaissance de la parole.

Guillaume Gravier(1) *François Yvon*(2) *Bruno Jacob*(3) *Frédéric Bimbot*(1)

(1) IRISA/INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex

(2) ENST Paris, Dpt. INFRES
46 rue Barrault
75634 Paris Cedex 13

(3) LIUM
Université du Maine
72085 Le Mans Cedex 9

Mél: ggravier@irisa.fr, yvon@enst.fr, bruno_jacob@lium.univ-lemans.fr, bimbot@irisa.fr

ABSTRACT

The *Sirocco* project aims at developing and distributing, under a free license, a speech recognition software toolkit according to open source standards. We present in this paper the main objectives of the project and describe the solutions implemented. In particular, *Sirocco* enables the use of contextual constraints on the pronunciation variants in the first decoding pass. We present preliminary results on the use of contextual transcription rules on a read speech transcription task.

1. INTRODUCTION

Le projet *Sirocco*[1] a pour objectif de développer et de maintenir une plate-forme rapide, performante et ouverte pour les systèmes de reconnaissance de la parole continue 'grand vocabulaire' (> 10k mots). Cette plate-forme se positionne comme un outil complémentaire aux divers outils aujourd'hui disponibles dans le domaine public, avec pour but de diffuser auprès du plus grand nombre d'équipes les technologies actuelles de reconnaissance vocale. Ce projet s'est développé de manière collaborative autour des partenaires suivants: l'Ecole Nationale Supérieure des Télécommunications, l'Institut de Recherche en Informatique et Systèmes Aléatoires, l'Institut de Recherche en Informatique de Toulouse, le Laboratoire d'Informatique d'Avignon, et le Laboratoire Lorrain de Recherche en Informatique. Le projet a bénéficié du soutien de l'INRIA dans le cadre d'une Action de Recherche Coopérative en 2000/2001.

La partie centrale de la plate-forme consiste en un décodeur de parole implémentant les algorithmes de l'état de l'art en matière de reconnaissance vocale grand vocabulaire. Ce décodeur est complété par un ensemble d'utilitaires dédiés à la mise en place des ressources linguistiques et acoustiques nécessaires à la reconnaissance. La plate-forme actuelle a été évaluée extensivement sur la tâche de dictée vocale définie dans le cadre des campagnes d'évaluation de l'AUPELF-UREF [3]. Elle s'accompagne d'un ensemble complet de documentations destinées à la fois aux utilisateurs et aux développeurs, et a été déployée avec succès chez les différents partenaires du projet.

Dans cet article, nous présentons les principales fonctionnalités de la plate-forme, en insistant tout particulièrement sur les algorithmes de décodage et de réévaluation. Nous présentons ensuite des résultats obtenus à l'aide des ressources publiquement distribuées lors des campagnes d'évaluation de l'AUPELF. La section suivante s'attarde sur une des originalités de notre décodeur, à savoir sa capacité à intégrer des contraintes explicites sur les séquences de transcriptions phonétiques susceptibles d'être reconnues, et présente des résultats d'expériences destinées à évaluer les apports de cette extension. Nous présentons pour finir un état des développements en cours et à venir.

2. LE SYSTÈME SIROCCO

Dans cette section, nous décrivons sommairement l'architecture générale de la plate forme, avant de présenter plus en détail les algorithmes de recherche utilisés pour la première passe de décodage, essentiellement inspirés de [7, 2], et pour la réévaluation linguistique. Pour illustrer les fonctionnalités du décodeur, nous donnons ensuite quelques résultats obtenus sur le corpus BREF [5].

2.1. Architecture du système

La plate-forme *Sirocco* est organisée de manière modulaire autour d'un décodeur implémentant une stratégie de recherche heuristique de la meilleure séquence de mot. Le décodeur permettant d'effectuer le décodage en plusieurs passes, la plate-forme intègre également un outil permettant de visualiser et de manipuler la sortie du décodeur sous la forme de graphes de mots. Ces deux composants sont présentés en détail dans les sections qui suivent. Autour de ce décodeur, la plate-forme intègre un ensemble d'utilitaires destinés à préparer les diverses ressources nécessaires à la mise en œuvre du moteur de reconnaissance. Ces utilitaires permettent de compiler les ressources nécessaires (arbre lexical, HMM, ML, ...) dans une représentation binaire directement exploitable par le décodeur. Pour chacune de ces ressources, nous avons défini des formats génériques et compatibles avec les standards existants. Ceci est en phase avec le souhait de positionner *Sirocco* comme un outil complémentaire des logiciels publiquement disponibles, que ce soit en matière de traitement du signal, d'apprentissage acoustique [9] ou d'apprentissage de modèles de langage [1].

2.2. Algorithme de recherche

Le décodeur *Sirocco* se base sur un algorithme de Viterbi utilisant une représentation arborescente des prononciations et un modèle de langage (ML) bigramme. Les techniques classiques d'élagage sont utilisées pour contrôler la taille de l'espace de recherche. L'algorithme utilisé suit principalement la description donnée dans [7]. Nous rap-

[1] Voir http://www.enst.fr/~sirocco

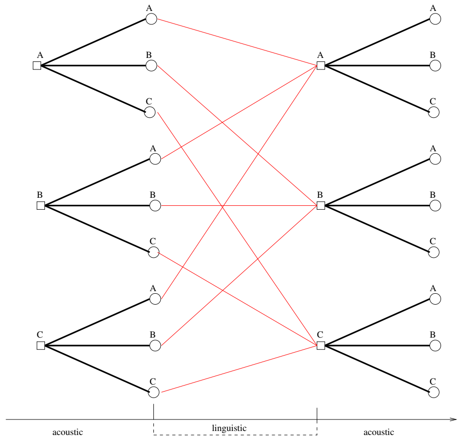

Figure 1: Propagation d'hypothèses dans un lexique organisé en arbre

pelons ici les fondements de l'algorithme en insistant sur le traitement des fins de mots. En effet, ce sont les fins de mot qui sont concernées par l'introduction de contraintes contextuelles comme nous le verrons au paragraphe 3.

Avec une représentation en arbre des prononciations, les fins de mots sont associées aux feuilles de l'arbre. En conséquence, l'application du ML bigramme n'est pas possible immédiatement lors de la transition entre deux mots puisqu'on ne connaît pas le mot suivant avant d'avoir atteint une feuille. Pour pouvoir appliquer correctement le ML, il convient de mémoriser le mot précédent, ce qui revient à considérer pour espace de recherche un graphe dans lequel il existe une copie de l'arbre lexical pour chaque mot (prédécesseur), comme illustré par la figure 1. Ainsi, à chaque feuille de l'arbre, il est possible d'appliquer le ML connaissant le mot précédent. Le retard dans l'application de la probabilité bigramme peut être partiellement compensé par l'application de techniques de "*look-ahead*" linguistique.

A l'intérieur d'une copie d'arbre, le décodage est basé sur le principe de programmation dynamique (DP) avec un élagage des hypothèses peu "*prometteuses*". A la fin d'un mot, la probabilité du ML est ajoutée à la vraisemblance du chemin et la maximisation est effectuée par rapport à l'ensemble des mots précédents. De manière plus formelle, en notant S_w la feuille correspondant à la fin du mot w, $h_w(t)$ le score de la meilleure séquence de mots terminant par w à t et $q_v(t, S_w)$ le score du meilleur chemin menant à l'état S_w à t dans l'arbre préfixé par le mot v, la maximisation est donnée par

$$h_w(t) = \max_{v \in \mathcal{V}} q_v(t, S_w) + \beta \ln P[W|v] \quad . \qquad (1)$$

Afin de pouvoir retracer la meilleure séquence de mots à l'issue du décodage, il est nécessaire de mémoriser le mot $\hat{v}$ qui maximise (1). Après cette maximisation, la recherche se poursuit en construisant des hypothèses ayant $\hat{v}$ pour mot précédent. Lors du traitement des fins de mots, il est également possible de mémoriser l'ensemble des mots utilisés dans la maximisation (1), pour pouvoir construire un graphe de mots à l'issue du décodage.

2.3. *Réévaluation des graphes de mots*

L'ensemble d'outils développé au cours du projet inclut un utilitaire destiné à la manipulation des graphes de mots produits par la première passe de décodage. Cet utilitaire permet d'exporter les graphes de mots vers des formats utilisés par d'autres logiciels. L'exportation de graphe de mots permet, par exemple, d'extraire des graphes au format FSM [6], permettant ainsi de leur faire subir divers opérations classiques (déterminisation, minimisation, extraction des N meilleurs chemins...). D'autres formats comme DOT (visualisation de graphe) et SLF (format HTK) sont également supportés.

Cet utilitaire permet également de réévaluer les chemins contenus dans le graphe à l'aide de modèles linguistiques plus précis que ceux utilisés pendant la première phase de décodage, et de produire les N meilleurs chemins. L'implémentation actuelle se limite à l'utilisation d'un ML trigramme. Cependant, la réévaluation des chemins est effectuée par un algorithme général, qui calcule les plus courts chemins dans l'automate à états finis valués résultant de la composition du graphe de mots avec n'importe quel automate valué défini sur le même vocabulaire. Le cas où l'automate valué est un modèle de langage permet une réévaluation linguistique du graphe mais d'autres utilisations de cet algorithme sont également possibles, par exemple pour calculer la distance du graphe à une phrase de référence.

2.4. *Quelques résultats ...*

Pour illustrer l'influence de l'élagage des hypothèses acoustiques, nous donnons table 1 les taux de reconnaissance en fonction du nombre maximum d'hypothèses acoustiques sur les données de la campagne AUPELF-B1 [3]. La tâche considérée dans la campagne AUPELF-B1 est une tâche de reconnaissance grand vocabulaire de parole lue. Les données, issues du corpus BREF [5], sont des phrases extraites du journal *Le Monde*. Un corpus d'environ 300 phrases lues par 20 locuteurs est utilisé pour le test, tandis que l'intégralité du corpus d'apprentissage de BREF est utilisé pour l'estimation des paramètres de modèles de phones hors-contexte. Le vocabulaire est constitué des 20 000 mots les plus courants dans le corpus d'apprentissage du modèle de langage (l'intégralité des années 88 et 89 du journal Le Monde).

Avec des ressources légèrement différentes de celles utilisées dans l'expérience précédente, on obtient un taux de reconnaissance de 61.7% après la première passe et le *rescoring* du graphe de mot correspondant avec un ML trigramme donne un taux de reconnaissance de 69.0%. Le taux d'erreur dans le graphe, calculé en cherchant le meilleur alignement entre la transcription de référence et le graphe, est d'environ 12% dans cette expérience. L'utilisation du ML "*look-ahead*" permet d'obtenir un taux de reconnaissance de 62.7% comparé à 60% en son absence.

Nous avons également tester le système avec des phones contextuels à l'intérieur des mots. Le taux de reconnaissance est alors de 63.7% (61.5% avec des monophones) après la première passe et de 68.2% (66.7% avec des monophones) après réévaluation linguistique.

Table 1: Taux de reconnaissance en fonction du nombre maximum d'hypothèses.

5 000	10 000	20 000	40 000	100 000
55.7	57.6	58.7	59.1	59.8

3. Règles contextuelles de transcription

Cette partie expose l'introduction de règles contextuelles de transcription dans le décodeur. Nous décrivons tout d'abord notre variante de l'algorithme de décodage pour prendre en compte de telles règles [4]. Nous présentons ensuite quelques résultats sur le corpus BDLEX pour diverses règles contextuelles extraites de MHATLex [8].

3.1. Introduction des règles dans le décodeur

La première étape dans l'introduction de règles contextuelles consiste à associer des classes aux mots du lexique (par exemple, *enfants*=pluriel, voyelle initiale). Le contexte dans lequel une variante de prononciation donnée s'applique est défini en terme de combinaison logique sur les classes du mot précédent et du mot suivant. Par exemple, il est possible de contraindre la variante [lez] du mot *les* à s'insérer uniquement dans des contextes où le mot suivant appartient aux classes "pluriel" et "voyelle initiale". Les formats *Sirocco* permettent donc d'associer des classes aux mots et de définir des contextes pour les transcriptions phonétiques.

Au niveau du décodeur, les informations sur les contextes sont associées aux feuilles de l'arbre lexical et une hypothèse de fin de mot correspond alors à une prononciation dans un contexte particulier. En plus de la maximisation (1), il est donc nécessaire de s'assurer que les contextes droit et gauche de la prononciation sont vérifiés. Du fait de l'organisation en arbre du lexique, il n'est bien évidemment pas possible de vérifier le contexte droit de l'hypothèse puisque le mot suivant est encore inconnu. En revanche, il est possible de vérifier la compatibilité du contexte droit de la règle utilisée pour le mot précédent avec le mot courant et le contexte gauche de la règle courante avec le mot précédent. Les chemins correspondant aux hypothèses de fin de mot ne vérifiant pas les conditions de compatibilité sont simplement supprimés de la recherche.

En plus des vérifications de contextes, cette approche demande de modifier l'algorithme de recherche de manière à garder une trace de la règle de transcription utilisée pour le mot précédent. Ce mécanisme est implanté en associant à chaque hypothèse acoustique un pointeur sur la règle de transcription précédente le long du meilleur chemin. Notons que le mécanisme est une heuristique de recherche. En effet, en théorie, pour garantir l'optimalité de la recherche, les hypothèses acoustiques ayant pour prédécesseur le même mot obtenu avec deux règles de transcription différentes doivent être considérées comme différentes, puisqu'elles ne partagent pas le même futur. Dans notre approche, ces deux hypothèses sont fusionnées en une seule hypothèse dont la règle de transcription du mot précédent est celle portée par la meilleure des deux hypothèses. Il est cependant possible de respecter le principe d'optimalité du décodeur en introduisant une copie de l'arbre lexical par règle de transcription plutôt que par mot, cette dernière approche augmentant toutefois considérablement la taille de l'espace de recherche.

3.2. MHATLex et les contextes

Pour tester l'apport des contraintes contextuelles, nous avons utilisé les ressources MHATLex développées à l'IRIT. MHATLex contient deux niveaux de représentation des transcriptions phonétiques: une représentation abstraite, dite *phonotypique*, qui condense un ensemble de représentations *phonétiques* valides dans un contexte linguistique défini. Le passage d'un niveau à l'autre est opéré par application de règles de réécriture, les transcriptions dérivées héritant des contraintes contextuelles de leur ancêtre. Considérons par exemple le verbe *prendre*, représenté au niveau phonotypique par deux entrées: [pRa~dR], valide si le contexte droit débute par une voyelle (ou une semi-voyelle), et [pRa~(dR@)], qui demande un successeur à initiale consonantique. Cette seconde représentation condense plusieurs variantes phonétiques correspondant à diverses réécritures du groupe [(dR@)] en [d], [n] ou [dR@]. Les prononciations résultantes [pRa~d], [pRa~n], et [pRa~dR@] héritent toutes des contraintes contextuelles de [pRa~(dR@)]. MHATLex inclut également diverses informations morpho-syntaxiques (lemme associé à une forme graphique, partie du discours, genre, nombre...), desquelles nous avons dérivé divers jeux de transcriptions contextuelles, ainsi que des classes lexicales permettant de définir des contextes. Environ la moitié de ces classes se fonde sur des propriétés morpho-syntaxiques[2], l'autre moitié des classes encodant des propriétés phonologiques.

Pour contrôler plus finement l'effet de l'introduction de contraintes contextuelles, nous avons manuellement catégorisé les règles dérivant les représentations phonétiques en fonction du phénomène phonologique sous-jacent: ainsi la dérivation du groupe phonotypique [(I)] dans [mef(I)a~] en [i] est-elle catégorisée comme une diérèse, l'alternative [j] étant catégorisée comme une synérèse. Ceci a conduit à marquer chaque transcription phonétique par l'ensemble des phénomènes linguistiques impliqués dans sa dérivation. Une fois ce marquage construit, nous avons plus particulièrement étudié trois phénomènes agissant à la frontière de mots: la liaison, les élisions ou réalisations de e-muets, et la chute de consonnes liquides finales, qui tous conduisent à des transcriptions phonétiques dépendant de l'environnement linguistique dans lequel elles s'insèrent. A titre d'exemple, considérons la forme *montrent* du verbe *montrer*. MHATLex distingue pour cette entrée deux contextes:

- si le mot suivant est à initiale vocalique, une liaison peut optionnellement être réalisée, avec maintient du e-muet; l'alternative étant l'absence de liaison et la chute du schwa. Dans ce contexte, deux variantes sont donc possibles: [mO~tR@t], marquée 'LRE' (liaison, liquide et e-muet maintenus); et [mO~tR], marquée 'lRe' (seul le [R] est maintenu).
- si, en revanche, le successeur débute par une con-

[2]Dans la mesure toutefois où le modèle de langage utilisé dans nos expériences ne distingue pas les homographes, les assignations de classes morpho-syntaxiques sont souvent ambiguës, ce qui limite leur utilité: ainsi une forme telle que *tombe*, est-elle simultanément un nom et un verbe.

Table 2: Taux de reconnaissance après décodage et après rescoring linguistique du graphe de mot pour divers jeux de règles contextuelles. Le nombre maximum d'hypothèses actives est limité à 10 000.

	*	E	L	R	ELR	M
2g	61.5	60.8	58.3	61.0	58.0	57.8
3g	66.7	65.6	63.1	66.1	62.3	62.1

sonne, la liaison est impossible, mais le groupe [R@] peut optionnellement être tronqué, donnant lieu à deux nouvelles variantes contextuelles: [mO~tR@] (marquée 'REl') et [mO~t] (marquée 'rel').

Nous avons dérivé de MHATLex plusieurs lexiques, qui diffèrent uniquement par la manière dont ces contraintes sont activées lors du décodage. Ainsi les transcriptions du lexique E ne conservent-elles les contraintes originelles de MHATLex que si elles impliquent la réalisation ou l'élision d'un e-muet, les autres transcriptions pouvant être librement sélectionnées. Dans ce lexique, toutes les transcriptions de *montrent* conservent donc leur contexte originel, à l'inverse par exemple de celles de *les* ([lez] et [le]) qui deviennent toutes deux valides, indépendamment de la nature du mot successeur.

3.3. Quelques résultats...

La table 2 donne les taux de reconnaissance sur le corpus AUPELF pour les différents jeux de règles contextuelles. La colonne (*) correspond à un lexique sans règles contextuelles, ELR à la combinaison des lexiques E, L et R (cf. ci-dessus) tandis que M correspond à l'ensemble des règles contextuelles définies dans MHATLex. En dehors des contextes, les conditions expérimentales sont similaires à celles décrites au paragraphe 2.4. Les résultats montrent clairement que l'utilisation de contextes n'entraîne pas d'amélioration du taux de reconnaissance. Ce résultat doit cependant être nuancé par le fait que ces expériences ont été menées avec les ressources publiques P0 de la campagne AUPELF et des modèles acoustiques non-contextuels.

4. CONCLUSION

Les résultats présentés dans cet article, bien qu'obtenus avec des ressources simples, montrent l'avancement du projet *Sirocco* et illustrent les possibilités des outils développés à ce jour. Des ressources plus performantes sont cependant nécessaires pour poursuivre les travaux sur l'utilisation de règles contextuelles dans le décodeur.

Parallèlement aux tâches permanentes d'optimisation et de documentation du code, nous comptons ajouter dans un futur proche plusieurs fonctionnalités aux outils existants. En particulier, le décodeur dans son état actuel n'utilise aucune des techniques de "*look-ahead*" acoustique, pourtant connues pour améliorer sensiblement les performances. Les outils d'adaptation au locuteur et aux conditions d'enregistrement sont pour l'instant également absents du décodeur.

A moyen terme, nous souhaitons également ajouter aux outils actuels un algorithme de recherche plus générique permettant de réaliser un décodage basée sur une grammaire régulière exprimée sous forme de graphe. Un tel outil permettrait d'une part de diriger la recherche par une grammaire pour une tâche précise et, d'autre part, de réévaluer des graphes de mots à l'aide de modèles plus complexes. En particulier, un des avantages d'un tel outil réside dans sa capacité à prendre en compte plus aisément des phénomènes inter-mots par l'utilisation de modèles de phones contextuels, ces derniers n'étant pas aisément utilisables dans le cadre d'une organisation en arbre du lexique.

Nous espérons pour finir que le caractère ouvert du projet permettra à d'autres laboratoires de contribuer, avec leurs connaissances propres, à l'amélioration et à l'extension des outils *Sirocco*.

REMERCIEMENTS

Les auteurs remercient Dominique Fohr (LORIA), Frédéric Béchet et Pascal Nocera (LIA) pour des interactions fructueuses au sujet du développement de la plateforme, ainsi que Fabien Antoine pour sa contribution au développement de l'algorithme de réévaluation de graphes de mots. Nous tenons également à remercier Akim Demaille pour son expertise en matière de développements de logiciels ouverts, et Arnaud Dauchy pour sa participation à la mise en place des modèles acoustiques. Les auteurs expriment finalement leur gratitude envers Guy Pérennou et Martine de Calmes (IRIT) pour avoir mis à disposition MHATLex.

BIBLIOGRAPHIE

[1] P. R. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Eurospeech*, 1997.

[2] N. Deshmukh, A. Ganapathiraju, and J. Picone. Hierarchical search for large-vocabulary conversational speech recognition. *IEEE Signal Processing Magazine*, pages 84–107, September 1999.

[3] J.-M. Dolmazon, F. Bimbot, G. Adda, M. ElBèze, J. C. Caërou, J. Zeilinger, and M. Adda-Decker. Organisation de la première campagne AUPELF pour l'évaluation des systèmes de dictée vocale. In *Journées Scientifiques et Techniques du Réseau Francophone d'Ingénierie de la Langue de l'AUPELF-UREF*, pages 13–18, 1997.

[4] G. Gravier, F. Yvon, B. Jacob, and F. Bimbot. On the use of contextual phonological rules in large vocabulary speech recognition. In *Eurospeech*, 2001.

[5] L. F. Lamel, J.-L. Gauvain, and M. EskĽnazi. BREF, a large vocabulary spoken corpus for French. In *Eurospeech*, pages 505–508, 1991.

[6] M. Mohri, F. Pereira, and M. Riley. *FSM Library – General purpose finite-state machine software tools*. AT&T.

[7] S. Ortmanns and H. Ney. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, 11:43–72, 1997.

[8] G. Pérennou and M De Calmès. MHATLex: Lexical resources for modelling the french pronunciation. In *2nd International Conference on Language Resources and Evaluation*, volume 1, pages 257–264, 2000.

[9] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK book (for HTK Version 3.0)*.