LIRIC : LInguistic speech Recognition In Context

Richard Beaufort, Sophie Roekhaut, Alain Ruelle

MULTITEL ASBL
Avenue Copernic 1, 7000 Mons, Belgique
beaufort,roekhaut,ruelle@multitel.be
http ://www.multitel.be/TTS

ABSTRACT

This paper presents a speaker-independent continuous Speech Recognition system that combines Multi-Layer Perceptron and Finite-State Machines (FSMs). In this system, FSMs represent all linguistic constraints, like phonemic durations, multiple phonetization and language modeling. The targeted application is a touristic Question-Answering system about Mons, a Belgian city. The evaluation of the system shows the interest of both using FSMs and adding linguistic constraints in Speech Recognition.

1. INTRODUCTION

Les systèmes de reconnaissance vocale développés actuellement sont principalement orientés « grand vocabulaire, parole continue, dépendant du locuteur » ou « petit vocabulaire, mots isolés, indépendant du locuteur ». L'objectif de LiRIC est de proposer un système « moyen vocabulaire, parole continue, indépendant du locuteur ». L'intérêt de ce type de système est de permettre à l'utilisateur, sans entraînement préalable, de s'exprimer librement dans un contexte donné. La difficulté de ce type d'application est d'une part de modéliser un langage destiné à une application définie et d'autre part, de modéliser la phonétisation d'un mot, dont la prononciation dépend du locuteur et du contexte d'énonciation. A l'aide de règles linguistiques représentées sous forme de machines à états finis (FSMs), nous avons défini un système répondant à ces exigences.

Cet article donne un rapide aperçu des machines à états finis, avant de présenter l'architecture globale de LiRIC. Il détaille ensuite les modèles linguistiques développés.

2. MACHINES À ÉTATS FINIS

Les FSMs [9, 7] regroupent les automates (FSAs), les transducteurs (FSTs), et leurs versions pondérées (WFSAs et WFSTs). Un FSA définit un langage L_A sur un alphabet A donné, et détermine si une séquence de symboles définie sur A appartient ou non à L_A. Un FST $L_{A:B}$ met en relation deux langages L_A et L_B respectivement définis sur les alphabets A et B ; il traduit une séquence de symboles définie sur A en une séquence de symboles définie sur B. Les versions pondérées WFSAs et WFSTs permettent la résolution de problèmes du type *recherche du meilleur chemin* dans un graphe de solutions [10].

Les FSMs autorisent une représentation compacte de l'information [6] et acceptent des opérations telles que l'intersection [14], mais aussi la composition [8, 11]. La composition est une généralisation de l'intersection définie sur les transducteurs (FSTs et WFSTs). A partir de deux (W)FSTs $L_{A:B}$ et $L_{B:C}$, la composition $L_{A:B} \circ L_{B:C}$ calcule l'intersection des deux machines sur le langage L_B et construit le (W)FST $L_{A:C}$, qui traduit les séquences de symboles du langage L_A en séquences de symboles du langage L_C. L'un des avantages de la composition est qu'elle permet de modéliser des tâches complexes à partir de tâches beaucoup plus simples. Cette opération fait partie intégrante du processus de reconnaissance dans LiRIC.

Les FSMs constituent un cadre de travail idéal pour la reconnaissance de la parole. Dans LiRIC, nous utilisons notre propre bibliothèque FSM.

3. ARCHITECTURE GLOBALE DU SYSTÈME DE RECONNAISSANCE LIRIC

LiRIC est un système de reconnaissance de la parole continue indépendant du locuteur qui combine Multi Layer Perceptron (MLP) et FSMs.

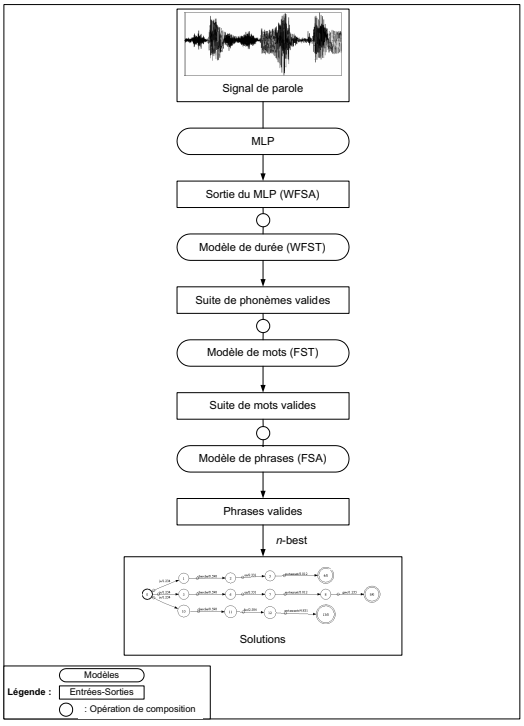

FIG. 1: Architecture globale du système hybride MLP/FSM LiRIC

Le système prend en entrée un signal de parole qui est

analysé par le MLP. La sortie du MLP, représentée sous la forme d'un FSM, est composée successivement avec les FSMs qui modélisent les différents stades de la reconnaissance. En sortie, le système propose les trois meilleures solutions en terme de phrases complètes ou de mots clés (cf. fig. 1).

3.1. Multi Layer Perceptron (MLP)

LiRIC intègre le MLP développé par le groupe Reconnaissance de la Parole de Multitel ASBL [3, 4]. Pour chaque trame de 30 ms de signal, le MLP émet 36 probabilités : 1 probabilité par phonème de la langue française.
De ces 36 probabilités, nous ne retenons que les *n* meilleures. Plus la valeur de *n* est élevée, plus le taux de reconnaissance augmente, mais plus le temps de calcul est important. Dans LiRIC, taux de reconnaissance et temps de calcul trouvent leur valeur optimale lorsque *n* vaut 15.

La sortie du MLP peut être représentée sous forme d'un WFSA A_{MLP} (cf. fig. 2).

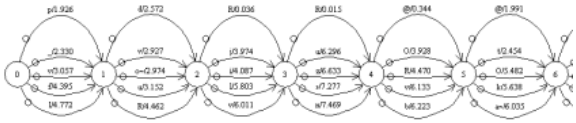

FIG. 2: Sortie du MLP (WFSA)

3.2. Modèle de durée

La durée d'un phonème varie selon plusieurs critères : son type (les voyelles sont généralement plus longues que les consonnes), le contexte dans lequel il apparaît et le locuteur qui le prononce. Sur base d'un corpus d'entraînement, nous avons calculé les probabilités de durée des différents phonèmes de la langue française. Notre modèle de durée est représenté sous la forme d'un WFST, dans lequel les durées sont définies en nombre de trames, étant donné que le MLP travaille sur des trames de 30 ms (cf. fig. 3).

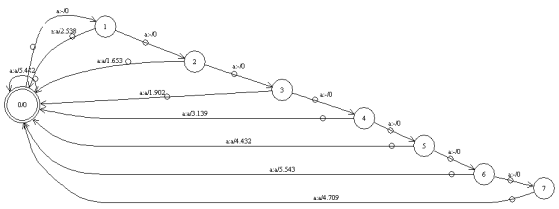

FIG. 3: WFST qui modélise la durée du phonème [a] en nombre de trames.

L'automate A_{MLP} est composé avec le modèle de durée. Le résultat est un treillis pondéré T_{PHO} de toutes les suites de phonèmes possibles pour le signal de parole d'entrée.

3.3. Modèle de mots

L'objectif du modèle de mots est de déterminer le ou les mots auxquels une séquence de phonèmes correspond. Ce modèle est représenté sous forme d'un FST qui prend en entrée l'ensemble des phonétisations valides pour chaque mot du langage et qui émet en sortie, pour chaque suite de phonèmes, le mot correspondant (cf. fig. 4). Les phonétisations multiples des mots ont été générées automatiquement à l'aide de règles linguistiques (cf. section 4.1).

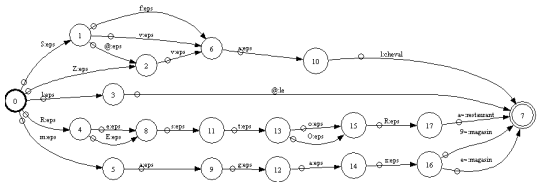

FIG. 4: T_{MOT} (FST)

La composition du treillis pondéré T_{PHO} avec le modèle de mots donne un treillis pondéré T_{MOT} de toutes les suites de mots valides pour le signal d'entrée.

3.4. Modèle de phrases

Le modèle de phrases représente, sous la forme d'un FSA, toutes les phrases acceptées par le langage de l'application (cf. fig. 5). L'ensemble des phrases du langage a été généré automatiquement à l'aide d'un compilateur d'expressions régulières (cf. section 4.2).

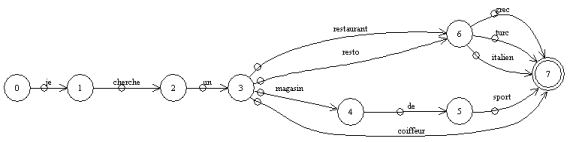

FIG. 5: Extrait du langage LiRIC (FSA)

La composition du treillis pondéré T_{MOT} avec le modèle de phrases donne un treillis pondéré $T_{PHRASES}$ de toutes les phrases valides pour le signal d'entrée.

3.5. Sélection des n-meilleures solutions

Au terme du processus, le système effectue sur $T_{PHRASES}$ une recherche des 3 meilleurs chemins soit en termes de phrases complètes soit en termes de mots-clés. La recherche en mots-clés permet d'augmenter le taux de reconnaissance (cf. section 5) en ne retenant de l'énoncé du locuteur que les mots dont le contenu est important pour le dialogue.

4. CONTRAINTES LINGUISTIQUES DU SYSTÈME LIRIC

4.1. Phonétisation multiple

En général, les systèmes de reconnaissance vocale proposent une phonétisation standard unique pour chaque mot du langage. Or, la prononciation d'un mot peut varier en fonction de différents paramètres tels que le contexte dans lequel le mot apparaît ou l'accent régional du locuteur qui prononce le mot. Il est donc intéressant d'autoriser plusieurs prononciations pour chaque mot du langage.

Il a été montré [2, 5, 13, 15, 12] que les variantes phonétiques au sein d'un mot et aux frontières du mot en contexte sont dues à des contraintes articulatoires que le locuteur relâche pour faciliter son élocution. Il est de ce

fait plus générique et plus concis de représenter les variantes phonétiques possibles en termes de modifications articulatoires. Par exemple, la règle « consonne sourde devient consonne sonore entre voyelles » pourra s'appliquer à toutes les consonnes sourdes ([p], [t], [k]…) dans n'importe quel contexte vocalique ([i], [e], [E]…). Les critères articulatoires qui définissent les phonèmes d'une langue sont :

– le type de phonème : voyelle, consonne, semi-voyelle
– le lieu d'articulation : bilabial, labio-dental, alvéolaire, palatal…
– le voisement : voisé, non-voisé
– la nasalité : nasal, oral
– les lèvres : arrondies, étirées
– le mode d'articulation (consonnes) : constrictif, occlusif, liquide
– l'aperture (voyelles) : 4 degrés, du plus proche au plus éloigné du palais

L'idée est donc de définir des règles qui ne concernent que les critères articulatoires nécessaires et suffisants. Ces règles sont rédigées sous la forme de règles de réécritures, compilées sous la forme d'un FST par un compilateur de règles de réécriture basé sur notre bibliothèque FSM [1].
Cependant, la phonétisation standard d'un mot dans le système est représentée en phonèmes. Pour cette raison, un premier FST convertit la séquence de phonèmes en séquence de critères articulatoires. Un second FST applique les règles articulatoires. Un troisième FST convertit les critères articulatoires en phonèmes. A la sortie de ces 3 FSTs, nous avons donc, pour un mot donné, toutes les phonétisations possibles décidées sur la base de critères articulatoires (cf. fig. 6).

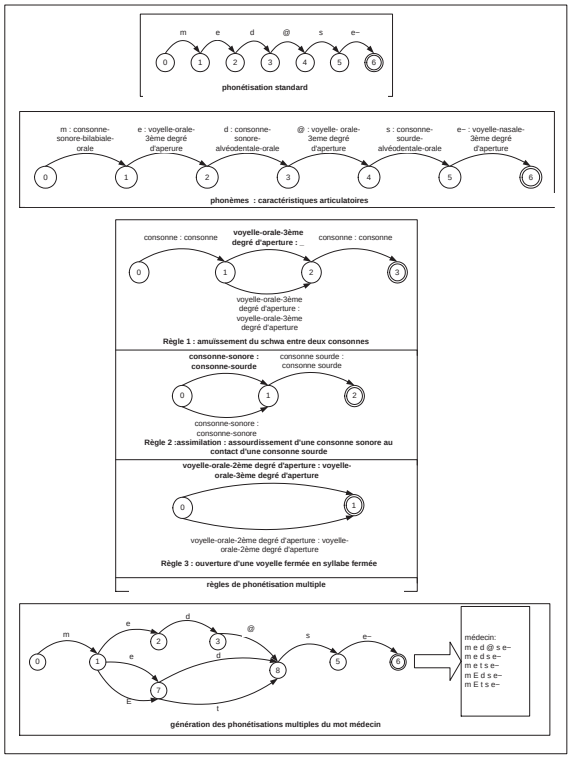

FIG. 6: Processus de génération de phonétisations multiples.

Notons que certains phénomènes articulatoires se produisent en frontière de mot et dépendent du contexte. Pour cette raison, notre système de phonétisation multiple travaille sur des phrases et non sur des mots isolés.

4.2. Définition du langage

L'application de démonstration basée sur LiRIC est un système de recherche d'informations touristiques sur la ville de Mons, en Belgique. Afin de définir le langage, nous avons dans un premier temps constitué une base de données des *focus* (restaurant, bar, magasin…) et des 300 *entités* de la ville (noms propres, par exemple « Alvolo » ou « Alain Afflelou »).
Dans un second temps, nous avons réalisé un recensement d'un large panel de formulations potentielles pour chaque type de question susceptible d'être posée. Ce recensement a été réalisé à l'aide d'un formulaire soumis à 50 personnes. La consigne du formulaire était : « Proposez toutes les formulations que vous pourriez employer pour obtenir de l'information sur... ». De manière à influencer le moins possible la formulation proposée, le formulaire donnait simplement une entité ou un focus, parfois accompagné d'une localisation (par exemple, « proximité Grand' Place », « Rue d'Enghien »…).

A partir de ce questionnaire, nous avons créé le lexique de l'application : un peu plus de 1000 mots. Nous avons ensuite dégagé 4 groupes « sémantiques » : l'entité, le focus, la question (je cherche, pouvez-vous m'indiquer…), et la formule d'introduction ou de clôture (bonjour, s'il vous plaît…). Nous avons ensuite décrit le langage de l'application sous la forme d'expressions régulières régissant l'agencement syntaxique des différents groupes sémantiques relevés (cf. fig. 7).

DET_DEF_MSC_SG	le
DET_DEF_FEM_SG	la
DET_IND_MSC_SG	un
DET_IND_MSC_PL	des
RESTO_NOM_MSC_SG	alvolo | bosphore | (brasse temps) | pastissou | (casse croûte)
RESTO_NOM_FEM_SG	sultane | (boule de bleu) | apidistra | cervoise
RESTO_CAT_MSC_SG	resto | restaurant
RESTO_CAT_MSC_PL	restos | restaurants
RESTO_NOM_MSC_SG_DEF	<DET_DEF_MSC_SG> <RESTO_NOM_MSC_SG>
RESTO_NOM_FEM_SG_DEF	<DET_DEF_FEM_SG> <RESTO_NOM_FEM_SG>
RESTO_CAT_MSC_SG_DEF	<DET_DEF_MSC_SG> <RESTO_CAT_MSC_SG>
RESTO_CAT_MSC_SG_IND	<DET_IND_MSC_SG> <RESTO_CAT_MSC_SG>
RESTO_CAT_MSC_PL_IND	<DET_IND_MSC_PL> <RESTO_CAT_MSC_PL>
RESTO_CAT_MSC_SG_NOM_MSC_SG_DEF	<RESTO_CAT_MSC_SG_DEF> <RESTO_NOM_MSC_SG_DEF>
RESTO_CAT_MSC_SG_NOM_FEM_SG_DEF	<RESTO_CAT_MSC_SG_DEF> <RESTO_NOM_FEM_SG_DEF>
INTRO	je (cherche | (voudrais trouver))
PHRASE	<INTRO> (<RESTO_CAT_MSC_SG_NOM_MSC_SG_DEF> | <RESTO_CAT_MSC_SG_NOM_FEM_SG_DEF>| <RESTO_CAT_MSC_SG_DEF>| <RESTO_CAT_MSC_SG_IND> | <RESTO_CAT_MSC_PL_IND>)

FIG. 7: modélisation d'une requête. La requête présentée concerne les restaurants de la ville de Mons.

A l'aide de notre compilateur de règles de réécriture [1], nous avons généré le FSA modélisant ce langage (cf. fig. 8), soit 116 millions de phrases différentes.

5. EVALUATION DU SYSTÈME LIRIC

Nous avons créé un corpus de données de 200 phrases appartenant au langage de l'application. 10 locuteurs différents ont procédé à l'enregistrement de ce corpus en milieu non bruité, ce qui donne un total de 2000 enregistrements. Un enregistrement est considéré comme reconnu lorsque la solution se trouve dans les trois meilleurs chemins proposés par le système.

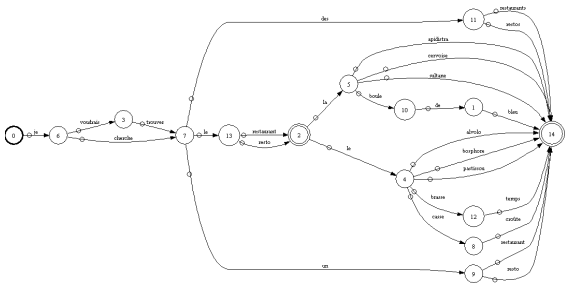

FIG. 8: machine représentant le langage décrit à la fig. 7

Cette évaluation porte essentiellement sur deux axes :
– reconnaissance de phrases complètes / mots-clés
– reconnaissance avec phonétisation multiple / sans phonétisation multiple

TAB. 1: LiRIC : taux de reconnaissance

	Total	**93,61**	**90,94**	**88,17**	**84,72**
%	1re sol.	89,44	81,83	83,67	77,22
	2e sol.	3,44	7,00	3,83	5,72
	3e sol.	0,83	2,11	0,67	1,67
		Mots-clés	Phrases	Mots-clés	Phrases
		Avec pho. multiple		Sans pho. multiple	

Les résultats obtenus montrent l'efficacité du système et l'intérêt de l'introduction du modèle de phonétisation multiple qui améliore de plus de 5% le taux de reconnaissance. La recherche en mots-clés permet de récupérer plus de 3% des erreurs.

Le temps de traitement dépend à la fois de la longueur de la phrase et de la taille du langage. Sous Linux et sur une architecture Intel Pentium 4 3.0 GHz pourvu d'1 Go de RAM, le système est en moyenne 11,7 fois temps réel pour le langage défini dans LiRIC.

6. CONCLUSION ET PERSPECTIVES

Nous avons présenté un système de reconnaissance vocale hybride MLP/FSM qui intègre des contraintes linguistiques et utilise un modèle de phonétisation multiple. Les premiers résultats obtenus sur l'application-test que nous avons présentée nous encouragent à poursuivre les recherches dans cette direction.

La mise au point de ce système a montré l'intérêt de la reconnaissance vocale par composition de machines à états finis. En effet, les FSM, outre leur efficacité, ont l'avantage de permettre la décomposition d'un problème complexe en sous-problèmes qui peuvent être traités indépendamment.

Afin d'augmenter les performances du système, nous envisageons de représenter les langages sous forme de modèles *n*-grammes pour élargir le langage. Nous comptons en outre développer un système de détection d'erreurs qui conduirait l'utilisateur dans un mode de dialogue parallèle en cas de détection d'erreurs partielles ou totales.

Le système a été facilement adapté à l'anglais pour le développement d'un scénario de déclenchement de catastrophe pétrolière et nous envisageons également son adaptation pour l'aide à la transcription de rapports médicaux.

7. REMERCIEMENTS

Nous tenons à remercier Laurent Couvreur, Jo Delafonteyne et Christophe Ris, qui nous ont aidé à intégrer le MLP.

RÉFÉRENCES

[1] R. Beaufort. *Parse & Merge* : compilation de Règles de Réécriture en transducteurs à états finis. In *Soumis à JEP*, Dinard, France, 2006.

[2] M. Debrock and P. Mertens. *Phonétique générale et française, Une introduction*. Presses Universitaires de Louvain, 1993.

[3] O. Deroo, H. Leich, J.M. Boite, S. Dupont, C Ris, and V. Fontaine. Hybrid HMM/ANN systems for for Speaker Independent Continuous Speech Recognition. In *Proceedings of ProRisc 8th Annual Work-Shop on Circuits, System and Signal Processing*, pages 137–141, 1996.

[4] S. Dupont, H. Bourlard, O. Deroo, V. Fontaine, and J.M. Boite. Hybrid HMM/ANN Systems For Training Independent Tasks : Experiments On Phonebook And Related Improvements. In *Proceedings of ICASSP*, pages 1767–1770, 1997.

[5] P. Hambye, M. Francard, and A.C. Simon. Phonologie du français en Belgique. Bilan et perspectives. In *La Tribune internationale des langues vivantes 33 La prononciation du français dans sa variation*, pages 56–63, 2003.

[6] M. Mohri. Compact representations by finite-state transducers. In *Meeting of the Association for Computational Linguistics*, pages 204–209, 1994.

[7] M. Mohri. Finite-State Transducers in Language and Speech Processing. volume 23, pages 269–311, 1997.

[8] M. Mohri, F. Pereira, and M. Riley. Weighted automata in text and speech processing. In *ECAI'96 Workshop*, pages 46–50, 1996.

[9] M. Mohri, F. Pereira, and M. Riley. A Rational Design for a Weighted Finite-State Transducer Library. In *Workshop on Implementing Automata*, pages 144–158, 1997.

[10] M. Mohri and M. Riley. An efficient algorithm for the *n*-best-strings problem. In *Proceedings of ICSLP'02*, 2002.

[11] F. Pereira and M. Riley. Speech Recognition by Composition of Weighted Finite Automata. In *Finite-State Language Processing*, pages 431–453, 1997.

[12] J.M. Pierret. *Phonétique historique du français et notions de phonétique générale. Nouvelle édition*. Peeters, Louvain-la-Neuve, 1994.

[13] J. Pohl. *Quelques caractéristiques de la phonologie du français en Belgique*, volume 60. 1983.

[14] E. Roche and Y. Schabes, editors. *Finite-State Language Processing*. MIT Press, Cambridge, Massachusetts, 1997.

[15] L. Warnant. *Dictionnaire de la prononciation française dans sa norme actuelle*. Duculot, Paris, 1985.