Mise en œuvre de LiaPhon pour la phonétisation avec variantes

Guide pour développeur

Ircam – Analyse-Synthèse

Christophe Veaux 12/10/2007

Introduction

Un phonétiseur est un élément indispensable pour tous les systèmes de traitement du langage parlé ainsi que pour la constitution des ressources nécessaires à l'analyse des corpus oraux. Nous présentons dans ce document le phonétiseur LiaPhon utilisé par le système d'alignement ircamAlign [REF] et pour la synthèse à partir du texte ircamTTS [REF].

LiaPhon a été développé par Frédéric Béchet au laboratoire d'informatique d'Avignon et est distribué sous licence GPL [REF]. En dehors d'être un logiciel libre, son principal intérêt est d'être basé sur une *approche par règles*. Cette approche rend la phonétisation aisément paramétrable et permet la génération de *variantes* de prononciation nécessaires pour l'alignement avec le signal de parole enregistré [REF].

Une description approfondie des principes de LiaPhon a été faite dans [REF]. Cependant nous avons procédé à certaines modifications en vue d'obtenir les variantes de phonétisation. D'autre part, si l'aspect paramétrable de LiaPhon constitue l'un de ses atouts, la syntaxe des règles et leur mise en œuvre n'est pas toujours explicite. Ce document vise donc à présenter la version modifiée de LiaPhon telle que nous l'utilisons et à apporter quelques éclaircissements concernant les règles.

Présentation

Le script à appeler pour effectuer la phonétisation standard de LiaPhon est **lia_text2phon**.

Il regroupe une succession de scripts unix communiquant par pipe et correspondant dans l'ordre aux étapes de traitement suivantes :

- formatage du texte
- étiquetage
- phonétisation

Nous présentons brièvement chacune de ses étapes.

1) Formatage du texte

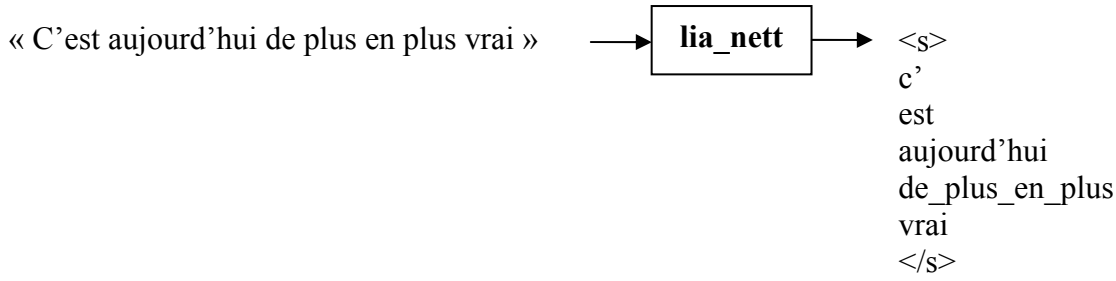
script associé : **lia_nett**

Le formatage du texte consiste à filtrer et segmenter le texte brut en entrée.

Le filtrage convertit les expressions non alphabétiques couramment utilisées dans les textes (chiffres, dates) en chaînes alphabétiques et élimine les caractères non autorisés. Cette étape doit donc éventuellement être adaptée selon le type de textes traités.

La segmentation découpe le texte en phrases puis en mots ou groupe de mots appelés « tokens ». Le texte en sortie comporte alors un token par ligne et des balises de début <s> et de fin </s> de phrases y sont insérées.

Exemple :



On notera que lors du découpage, LiaPhon cherche d'abord les expressions présentes dans le lexique de l'étiqueteur grammatical. Ceci lui permet de conserver les mots composés comme « aujourd'hui » sous forme d'entité unique. De même, les locutions adverbiales (« de plus en plus », « au fur et à mesure ») sont considérées comme un tout afin de faciliter l'étape d'étiquetage grammatical. C'est pourquoi on parle de segmentation en « tokens » plutôt qu'en mots.

fichiers utilisés : **lex10k** ou **lex80k** (lexique de l'étiqueteur)

2) Etiquetage

script associé : **lia_taggreac**

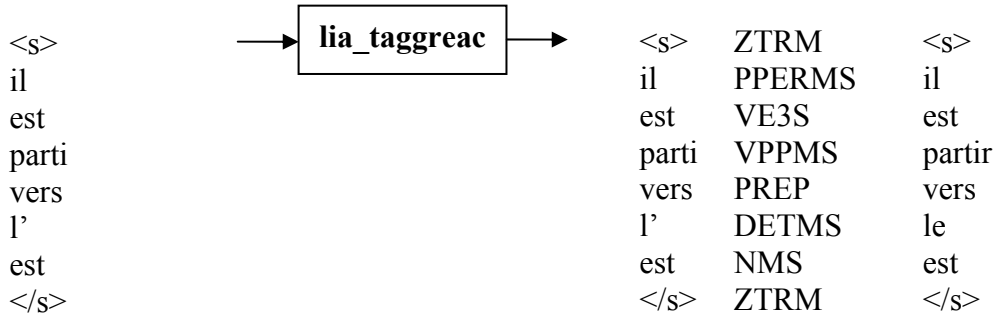
L'étiquetage attribue à chaque mot une *étiquette syntaxique* choisie parmi un jeu de 103 étiquettes (cf. annexe B). Cet étiquetage est utile à différents niveaux :

- La catégorie grammaticale permet généralement de lever l'ambiguïté sur la phonétisation des *homographes hétérophones* (par exemple, « il est parti vers l'est »).
- Les règles de liaison entre mots consécutifs dépendent de leur orthographe respective mais aussi de leur catégorie grammaticale.

- La prosodie est en partie conditionnée par la syntaxe, notamment les frontières des groupes accentuels.

L'étiquetage syntaxique effectué par LiaPhon n'est pas une analyse syntaxique complète car on se limite à identifier les catégories grammaticales des constituants de la phrase sans spécifier les relations entre ces constituants. Cette approche correspond à une *analyse syntaxique de surface*, elle a l'avantage d'être relativement robuste avec un taux d'erreurs qui reste cependant de l'ordre de 10%.

Exemple :



Les étiquettes grammaticales sont trouvées selon une *approche probabiliste* basée sur un modèle de langage. Plus précisément, on exploite un modèle tri-gram [REF CMU Statistical Language Toolkit] fournissant les probabilités de tous les triplets possibles d'étiquettes grammaticales consécutives dans une phrase. On utilise également un lexique fournissant la probabilité d'avoir tel ou tel mot pour une étiquette donnée. Ces connaissances permettent de se ramener au problème classique du *décodage d'une chaîne de Markov cachée* dont les états sont les *étiquettes grammaticales* et dont les observations sont les *mots*.

Si un mot n'est pas dans le lexique de référence, LiaPhon utilise un analyseur morphologique pour deviner la catégorie grammaticale du mot. Cet analyseur est également basé sur un modèle tri-gram mais cette fois-ci au niveau des lettres. Ainsi, à partir de tables fournissant la probabilité des triplets de lettres pour chaque étiquette grammaticale, on peut estimer l'étiquette la plus vraisemblable pour la séquence de lettres correspondant au mot en question.

LiaPhon possède également un analyseur de noms propres, lesquels sont identifiés par la présence d'une majuscule en début de mot. Cet analyseur est très similaire à l'analyseur morphologique (modèle tri-gram sur les lettres) et permet de calculer une probabilité d'appartenance à un groupe linguistique donné (8 langues étrangères sont modélisées).

Enfin, les expressions chiffrées et les sigles sont repérés par des étiquettes « sémantiques » (CHIF, SIGLE) qui conditionneront un traitement particulier de ces expressions par le phonétiseur. Cet étiquetage « sémantique » est effectué grâce à une base de règles prenant en compte le mot et son contexte d'occurrence dans la phrase. Ces règles sont très similaires aux règles de liaisons présentées ci-après.

<u>fichiers utilisés</u> :

- **lex10k** ou **lex80k** : lexique de mots avec leurs diverses étiquettes grammaticales possibles, les fréquence de chacune d'elles (relativement au corpus d'apprentissage) et les lemmes associés[1].
- **lm3classe.arpa** : modèle tri-gram des étiquettes grammaticales.
- **model_morpho** : modèle tri-gram des lettres pour l'analyseur morphologique décidant d'une catégorie grammaticale pour les mots hors lexique.
- **model_tri** : modèles tri-gram des lettres pour l'analyseur des noms propres décidant de l'appartenance à une langue.
- **regles_retik** : règles de ré-étiquetage sémantique pour les chiffres et sigles.

[1] Le lemme associé à un couple mot/étiquette est le mot de base dont dérive le mot en question lorsqu'il porte cette étiquette grammaticale.

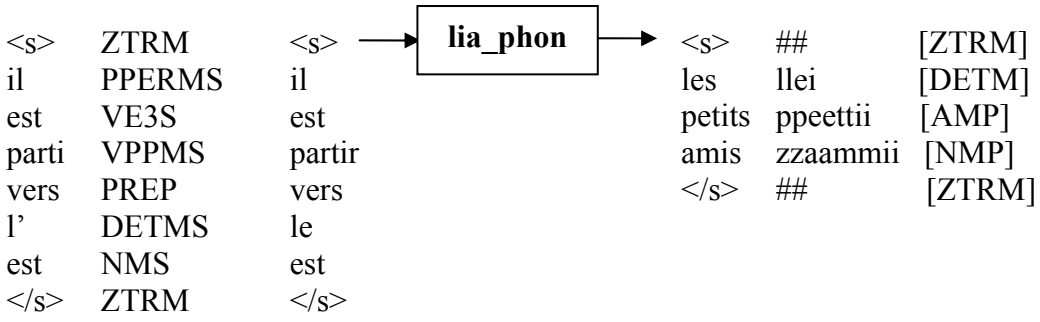
3) Phonétisation

script associé : **lia_phon**

Le module de phonétisation récupère le texte segmenté et étiqueté afin de le traduire en symboles phonétiques (alphabet LIA donné en Annexe A). Il procède en trois étapes :

- Décisions sur les liaisons à insérer entre les mots à partir de leurs graphies et de leurs étiquettes syntaxiques.
- Transcription phonétique des mots à partir de règles de phonétisation et d'un lexique d'exceptions.
- Post-traitement de la chaîne phonétique (gestion du e-muet, etc.)

Exemple :



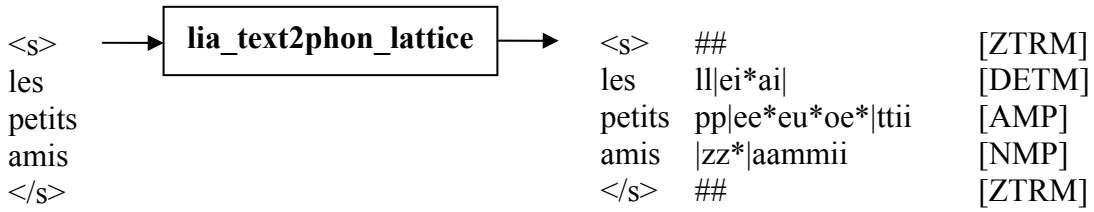
fichiers utilisés :

- **regles_l.pro3** : règles concernant les liaisons (obligatoires, interdites, facultatives).
- **h_aspi.sirlex** : lexique de mots commençant par un 'h' aspiré.
- **french01.pron** : règles de phonétisation standard pour le français.
- **list_exep** : lexique d'exceptions de phonétisation.
- **desigle.pron** : règles décidant si un sigle doit être épelé ou lu.
- **epeler_sig.pron** : règles de phonétisation des sigles épelés.
- **lire_sig.pron** : règles de phonétisation des sigles lus.
- **propername_[1..8].pron** : règles de phonétisation pour les noms propres de 8 langues différentes.
- **rule_phon.pro** : règles de post-traitement de la chaîne phonétique.

LiaPhon offre également la possibilité de phonétiser un lexique de mots (un mot par ligne) en proposant les variantes de prononciations. Le script associé est **lia_lex2phon_variante**, il ne peut cependant traiter des phrases. Aussi, nous avons apporté quelques modifications à LiaPhon de manière offrir la possibilité de générer des variantes de phonétisation sur une phrase donnée en entrée. Nous présentons cette fonctionnalité dans ce qui suit.

4) Génération de variantes

script associé : **lia_text2phon_lattice**



Par rapport au script de phonétisation standard (lia_text2phon), il a été nécessaire de remplacer ou d'introduire les fichiers de règles :

- **regles_liaison_en_contexte.pro3** : règles de liaisons (remplace règles_l.pro3).
- **french01_var.pron** : règles de phonétisation (remplace french01.pron).
- **rule_variante.pro** : règles de génération des variantes phonétiques *en contexte*.

Ajout ou modification de règles

Nous explicitons ici la syntaxe des différentes règles susceptibles d'être modifiées pour contrôler le mode de prononciation.

Remarques

Problèmes observés, todo list.

Exécutables et sources

Des liens vers les scripts **lia_nett**, **lia_text2phon** et **lia_text2phon_lattice** ont été placés dans /u/formes/share/bin/all-Linux. Ces scripts sont donc exécutables par défaut sur toutes les machines unix.

Pour toute modification des lexiques ou des fichiers de règles utilisés par LiaPhon, il est nécessaire de faire une copie locale des sources de LiaPhon par la commande **cvs co liaphon**. Suivre alors les instructions d'installation du fichier INSTALL à la racine de votre copie locale de LiaPhon.

N'oubliez pas de **recompiler les ressources** après toute modification des fichiers de lexique ou de règles (ou de modèles de langage s'il viennent à être modifiés).

Format ISO8859-1 absolument nécessaire en entrée (car format des fichiers de règles)

Annexe A Correspondance entre codes phonétiques LIA et XSampa

LIA	XSAMPA	EXAMPLES
ii	i	idiot, ami
ei	e	ému, été
ai	E	perdu, maison
aa	a	alarme, patte
oo	O	obstacle, corps
au	o	auditeur, beau
ou	u	coupable, loup
uu	y	punir, élu
EU	2	creuser, deux
oe	9	malheureux, peur
eu	@	petite, fortement
in	e~	peinture, matin
an	a~	vantardise, temps
on	o~	rondeur, bon
un	9~	lundi, brun
yy	j	piétiner, choyer
ww	w	quoi, fouine
pp	p	patte, repas, cap
tt	t	tête, net
kk	k	carte, écaille, bec
bb	b	bête, habile, robe
dd	d	dire, rondeur, chaud
gg	g	gauche, égal, bague
ff	f	feu, affiche, chef
ss	s	soeur, assez, passe
ch	S	chanter, machine, poche
vv	v	vent, inventer, rêve
zz	z	zéro, raisonner, rose
jj	Z	jardin, manger, piège
ll	l	long, élire, bal
rr	R	rond, charriot, sentir
mm	m	madame, aimer, pomme
nn	n	nous, punir, bonne
##	_	(silence marker)

Annexe B Etiquettes grammaticales utilisées par LiaPhon

Adverbes	ADV, ADVNE, ADVPAS
Adjectifs	AFS, AFP, AMS, AMP AINDFS, AINDFP, AINDMS, AINDMP
Déterminants	DETFS, DETFP, DETMS, DETMP DINTFS, DINTFP, DINTMS, DINTMP
Conjonctions	COCO, COSUB
Noms	NFS, NFP, NMS, NMP
Pronoms	PDEMFS, PDEMFP, PDEMMS, PDEMMP PINDFS, PINDFP, PINDMS, PINDMP PINTFS, PINTFP, PINTMS, PINTMP PPER1S, PPER1P, PPER2S, PPER2P, PPER3FS, PPER3FP, PPER3MS, PPER3MP PPOBJFS, PPOBJFP, PPOBJMS, PPOBJMP
Prépositions	PREFFS, PREFFP, PREFMS, PREFMP PRELFS, PRELFP, PRELMS, PRELMP PREP, PREPADE, PREPAU, PREPAUX, PREPDES, PREPDU
Verbes	V1S, V1P, V2S, V2P, V3S, V3P VA1S, VA1P, VA2S, VA2P, VA3S, VA3P, VAINF VE1S, VE1P, VE2S, VE2P, VE3S, VE3P, VEINF VINF VPPFS, VPPFP, VPPMS, VPPMP VPPRE
Noms propres	XFAMIL XPAYFS, XPAYFP XPAYMS, XPAYMP XPREF, XPREM XSOC, XVILLE
Ponctuations	YPFAI, YPFOR
Marqueurs de phrase	ZTRM
Mot hors lexique	MOTINC
Catégorie inconnue	<UNK>