



Institut de la francophonie pour l'informatique (IFI)

Bât. D, ruelle 42, rue Ta Quang Buu, Hanoï, Vietnam
<http://www.ifi.auf.org>

Travail Personnel Encadré
Systeme de transcription phonétique automatique
du texte en langue vietnamienne

Rapport final

Étudiant

NGUYEN Thi Minh Tuyen

Promotion 12, IFI

Superviseur

Eric Castelli (MICA)
Mathias Rossignol (MICA)
Alain Boucher (IFI)

Hanoï, juillet 2007

Remerciements

Ce travail a effectué dans le cadre d'un TPE (Travail Personnel Encadré).

Je tiens à remercier tout d'abord Monsieur Mathias Rossignol d'encadrer ce TPE avec enthousiasme et de me conseiller efficacement tout en me laissant travailler très librement.

Je tiens à remercier Monsieur Eric Castelli qui m'a apporté des connaissances de base sur la synthèse de la parole.

Je tiens à remercier tous les professeurs de l'IFI de m'équiper des connaissances nécessaires de haut niveau dans le domaine de l'informatique.

Je souhaite enfin remercier mes camarades de promotion 12 de l'IFI pour leur soutien et pour l'ambiance familière.

NGUYEN Thi Minh Tuyen
Juillet, 2007

Institut de la Francophonie pour l'Informatique, Vietnam

Table des matières

Remerciements.....	2
Table des matières	3
I. Introduction	5
II. État de l'art.....	6
1. eLite	6
2. LliaPhon.....	8
3. VNSpeech.....	10
4. Bilan.....	10
III. Alphabet Phonétique International(API).....	11
IV. Langue vietnamienne.....	12
1. Caractéristiques du vietnamien.....	12
2. Phonétique du vietnamien	12
3. Dialecte du Nord, dialecte du Centre et dialecte du Sud.....	16
V. Principe général du système développé	19
VI. Implémentation	21
1. Outils emplyés	21
2. Règles de transcription phonetique.....	22
3. Etapes de fonctionnement du systeme	23
VII. Résultats	28
VIII. Conclusion et perspectives	31

Liste des figures et tableaux

Tableau 1. Etapes nécessaires au fonctionnement d'un STP.....	11
Tableau 2. Consonnes et voyelles du vietnamien.....	13
Tableau 3. Premiers sons(un phonème – une écriture).....	14
Tableau 4. Premiers sons(un phonème – plusieurs écritures).....	15
Tableau 5. Sons noyaux.	16
Tableau 6. Derniers sons.....	16
Tableau 7. Changements de la prononciation du dialecte du Sud [4].	19
Figure 1. Traitement de la langue naturelle d'eLite	7
Figure 2. Structure de la syllabe vietnamienne.	13
Figure 3. Transcription en phonétique.	19
Figure 4. Transcription d'une syllabe en phonétique.....	20
Figure 5. Choix le fichier des règles.....	21
Figure 6. Sortir des résultats dans le fichier SSML à l'aide de TinyXML.....	22
Figure 7. Etapes de transcription phonétique.	23
Figure 8. Description des étapes.	23
Figure 9. Prétraitement.	24
Figure 10. Choisir la méthode de transcription phonétique.....	25
Figure 11. Transcription phonétique par l'unité de mot.....	26
Figure 12. Transcription phonétique d'une syllabe à l'aide des règles.....	26
Figure 13. Transcription phonétique lettre par lettre à l'aide des règles.....	27
Figure 14. Sortir des résultats dans le fichier.....	28
Figure 15. Fichier texte du résultat.....	29
Figure 16. Fichier SSML du résultat.....	30
Figure 17. Fichier texte du résultat avec le dialecte du Sud.....	31
Figure 18. Points d'amélioration possible pour notre travail.....	33

I. Introduction

Chaque jour, on communique pour s'informer, pour travailler ensemble par un moyen nommé la langue. Le développement des technologies génère de plus en plus de besoins pour la communication entre homme et machine ou pour la transmission d'informations dans les gares, les aéroports, *etc.* Le centre MICA, où se déroule mon travail de recherche, mène des recherches concernant le traitement de la langue vietnamienne (analyse et synthèse), pour lesquelles un système de transcription phonétique est un outil indispensable.

Un système de transcription phonétique (STP), qu'est-ce c'est ? Un STP, c'est un outil qui transforme un texte (ou un mot isolé) en une représentation de la manière dont il est prononcé (et non pas directement en parole). Ensuite, cette représentation peut être utilisée par un système de reconnaissance vocale ou de synthèse vocale. Typiquement, cette représentation utilise l'alphabet phonétique international qui est un standard international pour coder les sons de parole.

Dans le monde, il y a plusieurs STP construits pour plusieurs langues telles que l'anglais, le français, le hollandais, *etc.* Quant au vietnamien, il est assez simple de faire la connexion entre la parole et l'écriture parce que la correspondance entre écriture et prononciation est assez déterministe. En effet, le système d'écriture utilisé au Vietnam a été créé au 18^e siècle pour retranscrire précisément la phonétique de la langue. La création d'un logiciel de transcription phonétique automatique du texte en langue vietnamienne n'est donc pas trop difficile. Il y a des logiciels de synthèse de parole contenant un module de transcription telle que VnSpeech (Lê Hồng Minh 2004) [8], LacViet2002 (Lạc Việt 2002) [7], VVV (Lê Tang Hồ 2005) [10].

Mais en vietnamien, la prononciation dans chaque région est un peu différente, il n'y a pas une région où la prononciation est standard. Les variations régionales de prononciation sont donc assez importantes. La création d'un logiciel qui permet de produire, au choix, une prononciation pour un locuteur « standard » du Nord, du Centre, et du Sud est nécessaire.

Le but de ce TPE est réaliser un système de transcription du texte en langue vietnamienne avec l'identification et la quantification les variations de prononciation typiques du Nord, du Centre et du Sud du Vietnam, en s'appuyant sur des études linguistiques existantes ainsi que sur l'étude de données expérimentales, et d'intégrer ces connaissances a un STP classique afin de pouvoir produire, au choix, une phonétisation pour un locuteur "standard", du Nord, du Centre, ou du Sud.

Ce document est découpé en 7 parties : la section 2 est l'état de l'art. Dans cette partie, nous présentons quelques STPs existant pour la langue vietnamienne et pour d'autres langues. Ensuite, nous présentons l'Alphabet Phonétique International dans le section 3 et la langue vietnamienne dans le section 4. Dans le section 5, nous présentons le principe de fonctionnement de ce que nous avons fait. La section 6 est consacrée à la description technique. Dans la section 7, nous présentons le résultat de notre programme. Nous terminons avec la partie Conclusion et perspectives.

II. État de l'art

Comme nous l'avons dit dans la partie Introduction, il n'existe pas de STP de la langue vietnamienne indépendant. Dans des logiciels de synthèse de la parole comme VnSpeech, LacViet2002, VVV, il existe une fonction de transcription. Cela pose une question : Pourquoi on n'utilise pas un programme unique de transcription phonétique du texte vietnamien pour qu'on puisse l'utiliser dans plusieurs logiciels ?

Il y a encore une exigence assez importante, c'est la différence entre les dialectes dans trois régions : du Nord, du Centre et du Sud.

Aujourd'hui, l'alphabet phonétique international (API) est un moyen pour transcrire des langues, c'est un standard. Par rapport au vietnamien, on peut aussi utiliser l'API pour transcrire le texte. S'il existe un outil de transcription phonétique en API, il facilite de réaliser pas seulement la synthèse de la parole mais aussi d'autres applications, comme la reconnaissance vocale, pour faire le lien entre ce qui est entendu et les mots de la langue.

Il existe plusieurs outils qui réalisent la transcription phonétique pour plusieurs langues. Dans cette partie, nous présentons trois applications réalisant une transcription phonétique : eLite et Lliaphon pour les langues occidentales et VNSpeech pour le vietnamien.

1. eLite

Développé par Multitel, un centre de recherche dans le domaine du traitement de la parole, **eLite**[6,12](*Enhanced, Linguistically-based TExt-to-speech synthesizer*) est un système de synthèse de la parole complet pour deux langues : l'anglais et le français. Il fournit une plate-forme pour le développement et l'évaluation d'algorithmes intervenant en synthèse. Ce logiciel intègre Eliot pour le traitement du langage naturel (Natural Language Processing, NLP). Le traitement numérique du signal (Digital Signal Processing, DSP) est lui réalisé soit par le logiciel MBROLA, soit par la combinaison d'un sélecteur d'unités non-uniformes, LiONS, et d'un dérivé de MBROLA, TP-MBROLA.

La structure générale d'eLite se compose d'un noyau qui stocke des données, et de deux modules : le traitement de la langue naturelle (NLP) et le traitement du signal. NLP est le module que nous intéressent car il fournit une fonction de STP. Nous expliquons plus en détail les tâches de la partie NLP participant dans le processus de la transformation du texte en phonétique:

Preprocessing: Le rôle du préprocesseur est de diviser le texte d'entrée en mots, symboles et groupes de mots fortement liés, et de supprimer les caractères parasites (espaces, caractères de fin de ligne, caractères spéciaux...).

Analyse morphologique : L'analyse morphologique est basée sur la notion d'unité linguistique (une URL, un mot composé par exemple). Cette analyse s'effectue en deux étapes : identifier les différentes catégories syntaxiques

attribuables à chaque unité linguistique et attribuer une liste de natures lexicales possibles.

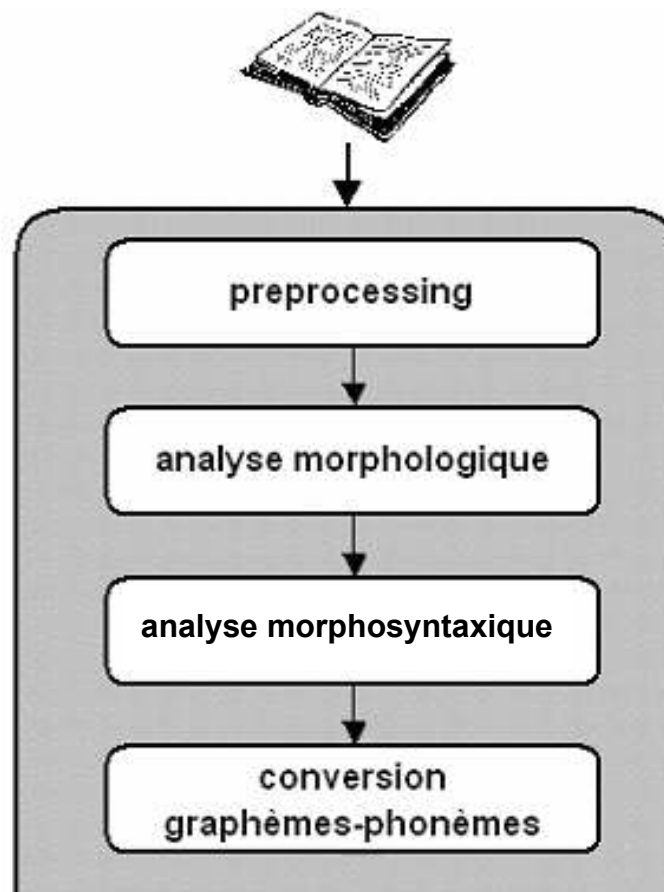


Figure 1. Traitement de la langue naturelle d'eLite

Analyse morphosyntaxique : Ce module a pour tâche de décider quelle catégorie morphosyntaxique correspond à chaque unité linguistique, étant donné le contexte de la phrase. Pour ce faire, l'analyseur, de type statistique, combine deux modèles. Le premier modèle est le modèle de langue, qui est classiquement un modèle trigramme. Le second modèle est le modèle de mot, qui est réintroduit dans le calcul grâce à l'emploi de classes d'ambiguïté.

Avec un STP, pourquoi a-t-on besoin de cette tâche ?

Un problème relativement particulier aux langues occidentales est celui des assez nombreux homographes hétérophones ("les poules du couvent couvent") dont l'ambiguïté ne peut être résolue en l'absence d'enregistrement témoin, alors qu'elle est cruciale en synthèse. Il est donc nécessaire d'ajouter le module d'analyse morpho-syntaxique pour résoudre ce problème.[2]

Conversion graphèmes-phonèmes : Ce module produit la transcription phonétique des mots du texte. Cette tâche contient deux étapes. Dans la première étape, cette transcription est générée indépendamment du contexte, à l'aide d'un arbre de décision compilé à partir d'un dictionnaire d'apprentissage. L'arbre détermine, pour chaque graphème, le phonème qui lui correspond, en tenant compte du contexte graphémique et de la nature lexicale du mot. Si la transcription générée est incorrecte pour la langue, le mot est alors épelé.

Dans la deuxième étape, la transcription du mot est remise en contexte. La prononciation d'un mot est en effet influencée par les mots qui l'entourent. Il est dès lors nécessaire d'examiner ces mots de manière à déterminer s'ils influent sur la prononciation du mot courant. Quelques phénomènes classiques sont la liaison (les_oiseaux), l'amuïssement (dix/ mille) et la lubrification du discours (comparez la prononciation de « e » à la fin de « *quelque* » prononcé dans « *quelque_personne* » mais pas dans « *quelque/ ami* »).

Dans la première étape, on utilise un arbre de décision compilé à partir d'un dictionnaire d'apprentissage. Pourquoi est-il nécessaire de faire comme ça ? Les nombreuses techniques qui sont utilisées pour générer la transcription peuvent être schématiquement classifiées en deux grandes familles : les approches par règles explicitement écrites ont généralement la faveur des experts linguistes et phonéticiens car elles sont plus facilement intelligibles, adaptables et contrôlables ; les approches statistiques basées sur l'apprentissage utilisant un lexique aligné de transcriptions présentent le grand avantage de requérir moins de travail expert (à la condition de disposer d'un lexique de transcriptions) et ainsi de se prêter plus facilement à l'extension à de nouvelles langues [2].

eLite supporte deux langues : l'anglais et le français. L'approche statistique avec la méthode de l'arbre de décision est donc la meilleure.

eLite est un bon logiciel supportant le STP. Il est flexible car il peut être étendu à plusieurs langues en utilisant la méthode de l'arbre de décision construit par apprentissage automatique.

2. LliaPhon

LliaPhon[11] est un système de phonétisation compact dérivé de LIA_PHON, développé sous licence GPL par le projet BigLux, sur la base de : LIA_PHON (Un système complet de phonétisation de textes[1]).

LliaPhon est un ensemble de modules écrits en C standard, en utilisant seulement les bibliothèques standard. Il compile et fonctionne parfaitement sous Linux.

Le but de LliaPhon est de produire un format phonétique qui facilite la transformation signaux audio. Actuellement, ce format est l'entrée pour l'algorithme MBROLA. LliaPhon utilise des étapes principales de Lia_Phon.

LliaPhon contient quatre opérations successives : *formatage*, *étiquetage*, *phonétisation* et *prosodie*. Le coeur du programme est une boucle qui lit une chaîne

de caractères, la transforme en une liste d'items successivement traitée par chacun des modules mettant en œuvre une des quatre opérations.

Dans les quatre opérations, les trois premières (formatage, étiquetage, phonétisation) sont nécessaires de construire un STP.

Formatage : Son objectif est de transformer en mots – en chaînes alphabétiques – toutes les expressions non alphabétiques couramment utilisées dans les textes : les chiffres, les dates, les numéros de téléphone, les adresses mail, etc.

Étiquetage : Le but est d'associer à chaque mot une étiquette grammaticale. C'est ici que l'on rencontre les données fondamentales nécessaires à la synthèse : d'abord le lexique – dont on a extrait les mots spéciaux et les locutions dans la phase précédente – qui contient la plupart des mots susceptibles d'être rencontrés avec leurs diverses étiquettes grammaticales possibles (le mot "voile" peut apparaître comme verbe ou comme nom ou encore comme adjectif s'il a été désaccentué) ainsi que d'autres informations associées aux couples mot/étiquette détaillés plus loin. Ensuite le catalogue qui est la liste des étiquettes grammaticales utilisées. Comme nous l'avons abordé dans eLite, cette opération est nécessaire à cause des nombreux homographes hétérophones.

Phonétisation : sert à associer à chaque mot une liste de phonèmes. C'est la partie importante pour un STP. Les phonèmes constituent les unités de prononciation.

Frédéric Béchet a introduit dans LiaPhon un codage systématique des phonèmes par 2 lettres, qui permet de simplifier le traitement informatique des caractères phonétiques. En effet, le support d'Unicode en C ne s'était pas encore imposé à l'époque. Par exemple, « on » au lieu de [õ] ou « uy » au lieu de [y] pour la prononciation de lui. C'est ce codage qui est utilisé dans LliaPhon et dans Lia_Phon.

Comme eLite, LliaPhon utilise des règles contextuelles – transcription d'un groupe de lettres en fonction des caractéristiques de celles qui les entourent – le lexique ne subsistant que pour les mots exceptionnels dont la prononciation échappe aux règles. Mais il y a une différence entre les deux logiciels. eLite utilise des règles qui sont créées automatiquement par l'apprentissage tandis que LliaPhon utilise des règles écrites à la main.

D'autres règles contextuelles, prenant comme unité élémentaire non plus un groupe de lettres mais un mot, sont utilisées pour décider de la présence ou non d'une liaison et de la façon de la prononcer.

La procédure de phonétisation d'une phrase passe par 3 phases :

- La détection des liaisons.
- La transcription phonétique des mots avec, en final, une étape de prise en compte des liaisons détectées.
- Traitements spécialisés post-transcription : traitement du 'e' muet et enchaînements vocaliques.

Dans LliaPhon, comme dans Lia_Phon, les liaisons sont détectées avant le traitement des mots eux-mêmes et leur effet sur la prononciation est ajouté après.

Ayant présenté deux exemples de STP pour langues occidentale, nous présentons maintenant le principe d'un logiciel qui supporte un STP pour le vietnamien, VNSpeech.

3. VNSpeech

VNSpeech[9] est un logiciel Text-to-speech pour le vietnamien basé sur des règles. Il est développé par Le Hong Minh, du Centre National pour le Progrès Technologique (*National Center For Technological Progress* en anglais). Ce logiciel utilise l'API pour la transcription phonétique avant la synthèse vocale. Le processus pour l'analyse des syllabes et la transcription phonétique se compose des étapes suivantes:

Etape 1: Traitement du codage du vietnamien (traiter chaque phrase)

- Déterminer le codage(TCVN3, VNI,UNICODE, ...)
- Transformer au codage intermédiaire (TCVN3)

Etape 2: Analyse d'une syllabe.

- Traiter le ton: est représenté par un chiffre (1-6)
- Traiter le premier son
- Traiter le son intercalé
- Traiter le son noyau
- Traiter le dernier son
- Pour chaque élément: le premier son, le son intercalé, le son noyau et le dernier son, trouver le phonème correspondant dans le tableau de chaque partie

Les tokens qui ne sont pas des mots vietnamiens sont traités indépendamment.

4. Bilan

Après avoir étudié les deux logiciel : eLite et LliaPhon, nous avons retiré quelques points importants. Pour construire un STP pour langue occidentale, avec deux logiciel ci-dessus, il existe des mêmes étapes principales : découpe en mots, analyse morphologique, analyse morphosyntaxique, transformation en phonétiques.

Pour le vietnamien, une langue monosyllabique, il est plus simple de construire un STP. Il n'y a pas d'homographes hétérophones, il n'est pas nécessaire d'ajouter le module Analyse morphosyntaxique. Découpe en mots, analyse morphologique et transformation en phonétiques sont les trois étapes principales pour le vietnamien. Pour l'étape de transformation en phonétique, comme des règles de transformation sont assez simples, la méthode utilisant des règles est adaptée. On peut résumer toutes les étapes dans le tableau 1.

	Elite	LliaPhon	Vietnamien en général
Découpe en mots	Oui	Oui	Oui
Analyse morphologique	Oui	Oui	Oui
Analyse morphosyntaxique	Oui	Oui	Non
Conversion graphèmes-phonèmes	Oui. Méthode : des règles contextuelles	Oui. Méthode : des règles contextuelles	Oui. Méthode : des règles contextuelles

Tableau 1. Etapes nécessaires au fonctionnement d'un STP.

Ayant présenté quelques travaux de référence, nous décrivons maintenant l'Alphabet Phonétique International pour avoir des connaissances de base sur la représentation de la prononciation et le contenu des fichiers produits par notre programme.

III. Alphabet Phonétique International(API)

La phonétique est une branche de la linguistique qui étudie les sons des langues. Elle étudie des sons (ou « phones »), la phonologie, elle, étudie comment ils sont agencés dans une langue pour former des énoncés. La transcription phonétique s'écrit entre crochets droits.

Nous donnons quelques exemples de l'API pour l'anglais, le français et le vietnamien.

Exemple :

Anglais : father ['fɑ:ðə], term [tɜ:m], alphabet ['ælfəbit],

Français : beau [bo], avoir [avwar], alphabet [alfabɛ]

Vietnamien: con [kɔn], bɔ [bɔ]

L'**alphabet phonétique international (API)** est un alphabet utilisé pour la transcription phonétique des sons du langage parlé. Contrairement aux nombreuses autres méthodes de transcription qui se limitent à des familles de langues, l'API est prévu pour couvrir l'ensemble des langues du monde. Développé par des phonéticiens britanniques et français sous les auspices de l'Association phonétique internationale, il a été publié en 1888. Sa dernière révision date de 2005. Le tableau de l'alphabet phonétique international est présenté sur le site web <http://www.arts.gla.ac.uk/ipa/> [14].

Aujourd'hui l'API devient un standard pour la transcription phonétique. Presque toutes les langues, dont le vietnamien, peuvent être transcrites en phonétique. C'est

la raison pour laquelle nous utilisons l'API pour la transcription phonétique dans notre TPE.

Nous présentons à la section suivante les caractéristiques linguistiques et phonétiques du vietnamien qui forment la base de notre travail.

IV. Langue vietnamienne

Avant de décrire la phonétique vietnamienne, nous présentons ici les caractéristiques du vietnamien et ses règles d'orthographe.

1. Caractéristiques du vietnamien

Le vietnamien est la langue isolante, c'est-à-dire qu'il est construit à partir de syllabes que l'on peut considérer isolément, et qui sont invariables. C'est la différence principale avec des langues comme l'anglais ou le français, qui sont des langues flexionnelles polysyllabiques. Cette spécificité influe tous les aspects de la langue.

Concernant la phonation, le vietnamien a une unité qui s'appelle «*tiếng* ». Quant à la syntaxe, cette unité est une syllabe. Le système du phonème vietnamien est diversifié et bien proportionné, donc le Vietnamien a capacité d'exprimer des unités significatives.

Par rapport aux caractéristiques du lexique : chaque syllabe a un sens. La syllabe est l'unité de base du système des unités significatives vietnamiennes. Avec la syllabe, on peut créer d'autres unités pour nommer des choses, des phénomènes, etc. par composition ou redoublement.

Concernant les caractéristiques de la syntaxe, le mot vietnamien ne change pas sa forme. Cette caractéristique va affecter d'autres caractéristiques de la syntaxe. L'ordre des mots dans une phrase est très important. Cela facilite la transformation phonétique. Avec le français et l'anglais, la forme du mot est assez importante [15].

2. Phonétique du vietnamien

Le vietnamien appartient au groupe Viet-Muong, branche Mon-Khmer de la famille Austroasiatique. Cette classification de l'origine du vietnamien est dérivée de l'hypothèse d'André-Georges Haudricourt, qui est, à l'heure actuelle, acceptée par le monde de la recherche. Selon A.G. Haudricourt, le vietnamien ressemblait originellement aux langues non toniques du groupe Mon-Khmer. Le caractère tonique du vietnamien fut ultérieurement ajouté grâce aux échanges culturels du voisinage avec le thaï. Le vietnamien avait déjà une riche littérature orale au moment de la conquête chinoise, vers le II^e siècle ap. J.C., puis, pendant une dizaine de siècles, sous la domination de la Chine, le chinois devint la langue administrative, et toutes les oeuvres « savantes » furent écrites en chinois. Le vietnamien a, durant cette période, été enrichi par un nombre important de mots chinois prononcés « à la vietnamienne » et appelés des mots sino-vietnamiens.

Au XII^e siècle apparaissent les caractères «nôm» (démotiques) basés sur l'écriture idéographique, qui permettent une transcription purement vietnamienne.

Au XVII^e siècle, le missionnaire jésuite français Alexandre de Rhodes met au point une romanisation de l'écriture vietnamienne, dite « quốc ngữ », toujours en usage, après être devenue officielle au XIX^e siècle en Indochine française. Sous la colonisation française, le vietnamien a également évolué par emprunt de mots et de constructions grammaticales françaises [3].

Le vietnamien possède 41 phonèmes, dont 23 consonnes, 13 voyelles simples, 3 diphtongues et 2 semi-voyelles, représentées par 29 lettres dans l'alphabet [3].

Voyelles	a ă â e ê i o ô ơ u ư y
Diphtongues	iê yê ia ya ươ ươ uô ua
Semi-voyelles	o u i y
Consonnes	b c d đ g h k l m n p q r s t v x
Combinaisons de consonnes	ch gh kh ng ngh nh ph th tr gi qu

Tableau 2. Consonnes et voyelles du vietnamien

Chaque syllabe vietnamienne se compose de 5 éléments : le ton, le premier son, le son intercalé, le son noyau et le dernier son [5]. Suivant leurs positions dans les syllabes, les lettres jouent des rôles différents. Selon ce rôle, il y a 5 systèmes phonétique: le système du premier son, le système du son intercalé, le système du son noyau, le système du dernier son et le système du ton. La structure de la syllabe est présentée à la Figure 2 :

Ton : sans accent(zéro),accent grave(`), accent retombant(?) accent remontant (~), accent aigu ('), accent intensif (.)			
	Rime		
<i>t</i>	<i>o</i>	<i>a</i>	<i>n</i>
Premier son [Consonne]	Son intercalé	Son noyau [voyelle]	Dernier son [Consonne/ semi-voyelle]

Figure 2. Structure de la syllabe vietnamienne.

Plus précisément, nous présentons chaque système.

Le système du ton

Le ton joue un rôle important dans une syllabe. Le ton distingue des syllabes à l'acuité. Il y a 5 accents représentés par 6 tons : sans accent (zéro), accent grave (`), accent retombant (ˊ), accent remontant (ˋ), accent aigu (ˆ), et accent intensif (˙)

Le système du premier son

Placé en première position, le premier son fait le commencement d'une syllabe. Les premiers sons sont toujours des consonnes.

Le vietnamien a au total 22 consonnes jouant le rôle le premier son. Ce sont /b, m, f, v, t, t', d, n, z, z', s, s', c, ç, ɲ, l, k, x, ɲ, ɣ, h, ʒ/. La plupart des phonèmes ont une représentation directe en un seule consonne :

Phonème	Ecriture	Exemple
/m/	m	mượt mà
/b/	b	buồn bã
/v/	v	vội vã
/f/	ph	phát phới
/t/	t	tan tác
/t'/	th	thơm tho
/d/	dd	đầy đà
/n/	n	no nề
/s/	x	xa xăm
/s'/	s	sớm sửa
/l/	l	long lanh
/c/	ch	chuồn chuồn
/ç/	tr	trục trặc
/ɲ/	nh	nhanh nhẹn
/x/	kh	khô khốc
/h/	h	hối hả
/z/	r	ra ruộng

Tableau 3. Premiers sons(un phonème – une écriture).

Il existe certains cas dans lesquels un phonème est représenté par deux ou trois consonnes combinés comme présenté dans le Tableau 4 :

Phonème	Écriture	Exemple
/z/	d/gi	da dẻ/gia đĩnh
/k/	q/c/k	kì, kẻn, quân, con cáy
/ɣ/	g/gh	ghi, ghé, gảy gổ
/ŋ/	ng/ngh	nghi, nghề, ngủ ngon

Tableau 4. Premiers sons(un phonème – plusieurs écritures).

Le système du son intercalé

Le son intercalé est le troisième élément, après le premier son. Il fait changer le timbre de la syllabe après le commencement, plus précisément, elle rend la syllabe plus grave. Le son intercalé se compose du phonème semi-voyelle /u/ et le phonème « zéro » (le phonème nul).

Par exemple :

loạt : cette semi-voyelle est présentée par la lettre o

đẹp, xinh : le son intercalé est le phonème zéro

Le système du son noyau

Le son noyau se place en quatrième position dans une syllabe, il est le noyau, le sommet de la syllabe, il porte le timbre principal de la syllabe. La voyelle joue le rôle de son noyau, elle n'est jamais absente dans une syllabe parce qu'elle porte l'intonation.

Le vietnamien a en tout 13 voyelles simples et 3 diphtongues. Ce sont /i, e, ɛ, ɛ̃, ɯ, ɤ, ɤ̃, a, ă, u, o, ɔ, ɔ̃, i_e, ɯ_ɤ, u_o/. Ils sont représentés en écriture dans le tableau 5 :

Phonème	Écriture	Exemple
/i/	i, y	im ìm, ý chí
/e/	ê	ê chẻ, êm đẻm
/ɛ/	e	e dẻ, e thẻn
/ɛ̃/	a(anh, ach)	anh ách, xanh xanh
/ɯ/	ư	từ từ, lử thứ
/u/	u	tu hú
/o/	ô	hồ đờ
/ɔ/	o, oo (oong, ooc)	co ro, xoong
/ɔ̃/	o (ong, oc)	vòng lọng
/ɤ/	ơ	bơ phờ
/ɤ̃/	â	ân cần

/a/	a	la ðà
/ă/	ă, a(au, ay)	ăn năn, rau ðay
/i_e/	iê, yê, ia, ya	kia kia, khuya, yêu kiều
/u_x/	ưƠ, ưA	lướt thướt, lưA thưA
/u_o/	uô, ua	tuốt tuồn tuột, tua rua

Tableau 5. Sons noyaux.

Le système du dernier son

Le dernier son se place à la fin de la syllabe, il termine une syllabe. Donc, quand il y a un dernier son, la syllabe n'a pas de capacité d'ajouter un phonème après lui. Les derniers sons peuvent être des consonnes, des semi-voyelles ou « zéro ». Le dernier son « zéro » est un phonème vide et il n'est pas représenté par l'écriture.

En dehors du phonème « zéro », le système du dernier son a 8 consonnes : /m, n, ŋ, p, t, k/ et 2 semi-voyelles /u_x, i_y/. Ils sont représentés en écriture dans le tableau 6 :

Phonème	Ecriture	Exemple
/m/	m	êm ðêm
/n/	n	lăn cãn
/ŋ/	ng, nh	lang thang, thình linh
/p/	p	pháp phới
/t/	t	túu tít
/k/	c, ch	tan tác, thích thú
/u_x/	u, o	rêu rao
/i_y/	i, y	gửi, nơi, ấy

Tableau 6. Derniers sons.

Comme nous savons, le dialecte est un grand problème du Vietnamien. Nous présentons dans la partie suivante la différence entre les trois dialectes : du Nord, du Centre et du Sud.

3. Dialecte du Nord, dialecte du Centre et dialecte du Sud.

Aujourd'hui, il n'existe pas de locuteur « standard ». On peut subdiviser le vietnamien en trois dialectes : du Nord, du Centre et du Sud.

Ces dialectes se distinguent par des variations de vocabulaire et une prononciation différente des lettres et digrammes.

Cependant, en pratique, chaque personne semble adopter la prononciation qui lui plaît, peu importe la région. Ces différences ne posent aucun problème de compréhension, la limite entre deux prononciations étant parfois imperceptible.

1. Le dialecte du Nord

Dans le dialecte du Nord, il y a deux systèmes qui sont différents des autres. Ce sont le système du ton et le système du premier son.

Le système du ton a 6 tons, comme le dialecte standard.

Le système du premier son a 20 consonnes. Dans ces 20 phonèmes, il y a pas de distinction entre s/x, r/d/gi, tr/ch.

Le dialecte de Ha noi et des régions voisines ((Hà Bắc (*Bắc Ninh, Bắc Giang*), Vĩnh Phú (*Vĩnh Phúc, Phú Thọ*), Hà Sơn Bình (*Hà Tây, Hoà Bình*), Hải Hưng (*Hải Dương, Hưng Yên*), Hải Phòng): c'est la région typique du dialecte du Nord

Dans les régions de l'aval de Song Hong et la zone littorale (Thái Bình, Hà Nam Ninh, Quảng Ninh), il existe la distinction entre d et gi, r; entre tr et ch qui n'existent pas dans d'autres dialectes du Nord.

2. Le dialecte du Centre

Dans le dialecte du Centre, le système du ton a seulement 5 tons. Il n'y a pas de différence de prononciation entre le ton retombant et le ton remontant.

Le système du premier son a 23 consonnes. Dans ces 23 consonnes, il y a 3 consonnes « coupant la langue » : /**ʃ, z, ʈ**/ (s, r, tr). Dans plusieurs patois, il existe 2 consonnes aspirées [ph, kh] au lieu de 2 consonnes fricatives /**f, x**/ dans le dialecte du Nord.

Le dialecte du Centre peut être divisé en 3 dialectes plus petits, la base de cette division est la différence du ton.

- Le dialecte de Thanh Hoa : ne distingue pas la prononciation entre le ton retombant et le ton remontant, les autres tons sont semblables à ceux du dialecte du Nord.

- Le dialecte de Nghe Tinh : ne distingue pas entre le ton retombant et remontant, mais les 5 tons forment un système qui est différent de celui du Nord parce que le ton est plus grave.

- Le dialecte de Binh Tri Thien (Quang Binh, Quang Tri, Hue) : ne distingue pas entre le ton retombant et le ton remontant. Le système du ton est semblable à celui de Nghe Tinh, mais la région de Hue a un système de la rime et du dernier son semblable à celui du Sud.

3. Le dialecte du Sud

Comme le dialecte du Centre, le dialecte du Sud a 5 tons, le ton retombant et le ton remontant ne sont pas distingués.

Le système du premier son : il y a 23 premiers sons. Il existe des consonnes « coupant la langue » /**ʃ, z, ʈ**/ (s, r, tr). Au Sud, on peut prononcer /r/. Mais le dialecte du Sud manque la consonne /v/, il existe /w/ et /z/ est remplacé par /j/. Le dialecte du

Sud n'existe pas plusieurs rimes en comparaison avec le dialecte du Nord et dialecte du Centre. Il manque /-ŋ, k/.

Le dialecte du Sud peut être divisé en 3 dialectes plus petits :

- Le dialecte Quang Nam – Quang Ngai : il existe un changement de /a/ et /ă/ en association avec les derniers sons.

- Le dialecte Quy Nhon – Thuan Hai : a des caractéristiques typiques du dialecte du Sud.

- Le dialecte méridional : il n'y a pas de différence entre

-in, -it avec -inh, -ich

-un, -ut avec -ung, -uc

Dans cette région, on confond s avec x et tr avec ch comme dans le dialecte du Nord.[13] Tous les changements de la prononciation dans le dialecte du Sud sont présentés dans le tableau 7 :

écriture	Standard	Dialect du Sud	exemple
ho-, hu-, qu-, « oa, uê, uy »	/h-/ , /k-/ , /ʔ/	/w/	hoán vị /waŋ jì/, huề vốn /we jɔŋ/, gỏi quà /ɣoi wa/, làm quen /lam wɛŋ/, oan úc/ waŋ uk/, uể oải/we wai/, uy quyền/wij wiŋ/
s-	/ʃ/	/s-/	sổ sách/so sɜt/
v-, d-, gi-	/v,z,z/	/j/	vui về/ju :j jɛ :/, giàu có/ja :w kɔ :/, dám làm/ja :m la :m/
-uy-	/-wi-/	/-i-/	thuỷ/t'ij/
-oa-	/-wa-/	/-a-/	toác hoác/ta :k wa :k/
-uyen-, -uya-	/-wiɣ-/	/-iɣ-/	thuyền/t'iɣŋ/
-wa-	/-wa :-/	/-a :-/	toà/ta :/, loa/la :/
-wâ-	/-wɣ-/	/-u-/	luân/luŋ/
-êu	/-e :w/	/-i :w/	cây nêu/kɣj ni :w/
-êm	/-e :m/	/-i :m/	thêm vào/t'i :m ja :w/
-êp	/-e :p/	/-i :p/	cơm nếp/kɣ :m ni :p/
-ên, -ênh	/-en/	/-ə :n/	bệnh vực/bə :n juɔk/
-êt, -êch	/-et/	/-ə :t/	chết chóc/cə :t cɔk/
-uôu, -uô	/-u :w/	/u/	con hươu/kɔ :ŋ hu/
-ai, -ay	/-a :j/ /aj :/	/-a :j/	làm mai/la :m ma :j/ nhà may/ɲa : ma :j/

-ao	/-a :w/	/-a :w/	tô cháo/to ca :w/
-au	/-aw :/		cháu nội/ca:w no:j/
-um	/-u :m/	/-u :m/	tùm lum/tu:m lu:m/
-up	/-u :p/	/u :p/	ẵ núp/ɤη: nu:p/
-ôm	/-o :m/	/ɤ :m/	nôm na/nɤ :m na :/
-óp	/-o :p/	/ɤ :p/	hôi hốp/ho :j hɤ :p/

Tableau 7. Changements de la prononciation du dialecte du Sud [4].

Nous terminons la partie théorique et continuons en présentant la partie pratique de notre TPE : le principe général du système que nous avons développé, puis son implementation.

V. Principe général du système développé

Comme nous l'avons présenté, pour transcrire le texte en phonétique, il existe de nombreuses techniques qui sont classées en deux grandes familles : les approches par règles et les approches statistiques basées sur l'apprentissage. Les approches par règles sont plus facilement intelligibles, adaptables et contrôlables. Les approches statistiques basées sur l'apprentissage d'un lexique aligné de transcriptions présentent le grand avantage de requérir moins de travail expert et d'ainsi se prêter plus facilement à l'extension à de nouvelles langues. Après avoir étudié les deux approches, nous avons décidé de choisir l'approche par règles, par trois raisons. Premièrement, dans notre TPE, nous avons travaillé seulement avec le vietnamien, on n'a pas besoin d'extension à nouvelles langues. Deuxièmement, des règles vietnamiennes sont assez simples et déterministes, utilisant des règles sont donc préférables. Troisièmement, nous ne disposons pas de lexique pour réaliser l'apprentissage, et en construire un a la main demanderait plus de travail que construire directement les règles.

Selon la partie théorique du vietnamien et la phonétique du vietnamien, nous vous présentons des principes généraux de notre travail.

Voici le diagramme pour transcrire en phonétique. A partir d'un fichier texte, on transforme ce fichier en phonétique et la sortie est un fichier phonétique.

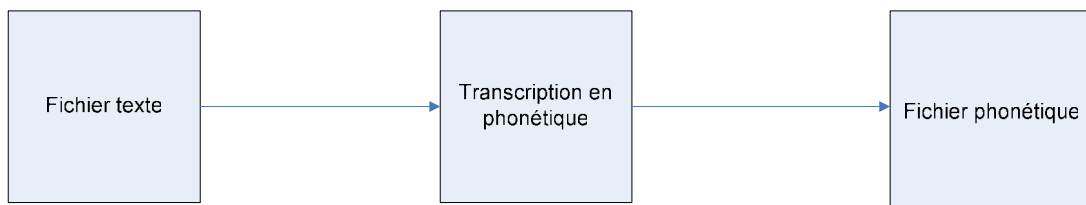


Figure 3. Transcription en phonétique.

Voici le diagramme pour découper un mot vietnamien en parties. Une syllabe est découpé en 5 parties : ton, premier son, son intercalé, son noyau et dernier son. L'API propose des symboles pour la représentation des tons [15], mais ceux-ci ne permettent pas de couvrir tous les tons vietnamiens. Dans cette partie, le ton est donc représenté par des chiffres (0 - 5) . Les parties: premier son, son intercalé, son noyau et dernier son sont représenté par l'API selon des règles disponibles et selon le dialecte de chaque région.

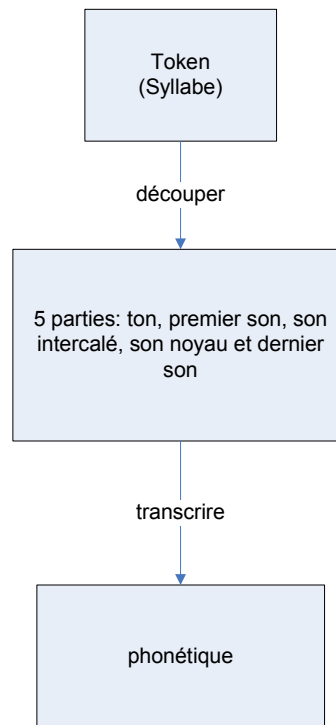


Figure 4. Transcription d'une syllabe en phonétique.

Dans ce cas si la structure du mot ne permet pas un découpage en 5 parties conforme aux règles d'orthographe vietnamiennes on a un « mot anormal ». Il y a deux solutions : s'il s'agit d'une abréviation classique, on peut utiliser un dictionnaire d'abréviations, sinon on transcrit lettre par lettre.

Il y a 4 choix pour des règles selon le dialecte : standard, du nord, du centre ou du sud. Des règles de transformation entre écriture et son phonème sont stockés dans des fichiers XML pour améliorer la flexibilité. C'est un grand avantage de notre travail puisque des règles ne sont pas dans le logiciel, on peut changer facilement sans compiler le programme encore une fois.

Dans Figure 5, on peut voir que le choix un des 5 fichiers des règles est assez facile. Si l'utilisateur veut changer une règle, il peut changer simplement en modifiant un de ces fichiers.

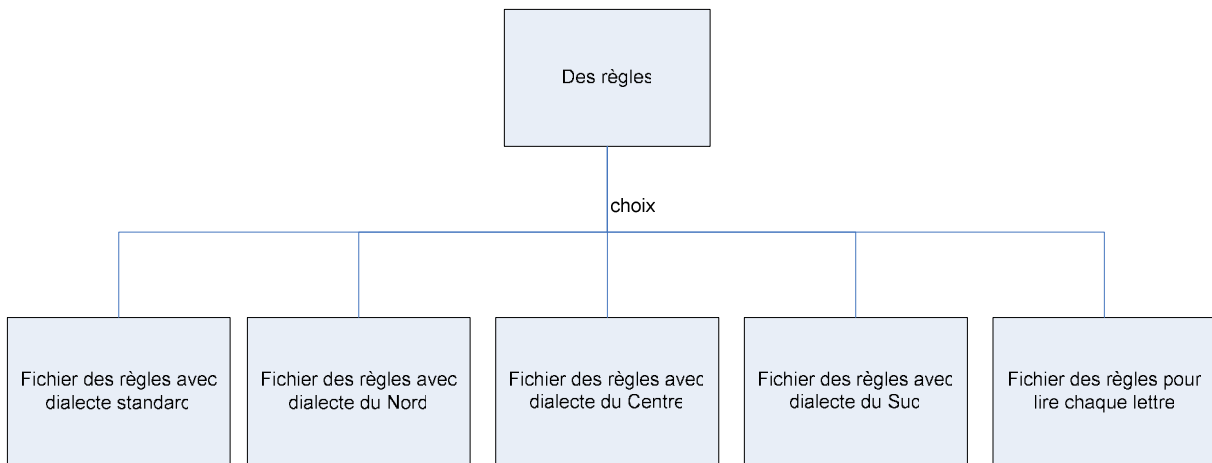


Figure 5. Choix le fichier des règles.

Le mode de fonctionnement détaillé est présenté dans la partie suivante.

VI. Implémentation

1. Outils employés

C++ standard

Le but du programme développé est afin d'assurer une portabilité optimale du code, et soigneusement structuré afin de pouvoir être intégré comme composant d'un système. Il est donc nécessaire d'utiliser C++ avec sa bibliothèque standard. La bibliothèque standard du C++ est une bibliothèque de classes et de fonctions standardisées pour le langage C++. Elle fournit un certain nombre de classes comme par exemple : la manipulation de chaînes de caractères, la manipulation de flux (fichier...), des conteneurs permettant par exemple de parcourir leur contenu, une collection d'algorithmes de bas niveau. Cette bibliothèque est le résultat de l'évolution de plusieurs bibliothèques, parfois développées indépendamment par plusieurs fournisseurs d'environnements C++, qui ont été fusionnées et normalisées afin de garantir la portabilité des programmes qui les utilisent.

TinyXML

Pour lire et sauvegarder les XML, nous utilisons un outil qui s'appelle TinyXML. TinyXML est un analyseur syntaxique petit et simple de XML qui peut être intégré facilement aux autres programmes. TinyXML analyse un document XML et le construit avec des opérations : lire, modifier et sauvegarder. Il existe des façons différentes pour l'interaction avec des données XML. TinyXML utilise DOM (Document Object Model), c'est-à-dire des données XML sont analysées en objets C++ qui peuvent être parcourus et manipulés et stockés directement en mémoire, ou être stockés en fichiers. TinyXML est conçu de manière proprement structurée. Il a 2 fichiers header et 4 fichiers cpp. TinyXML supporte des textes codés en UTF-8.

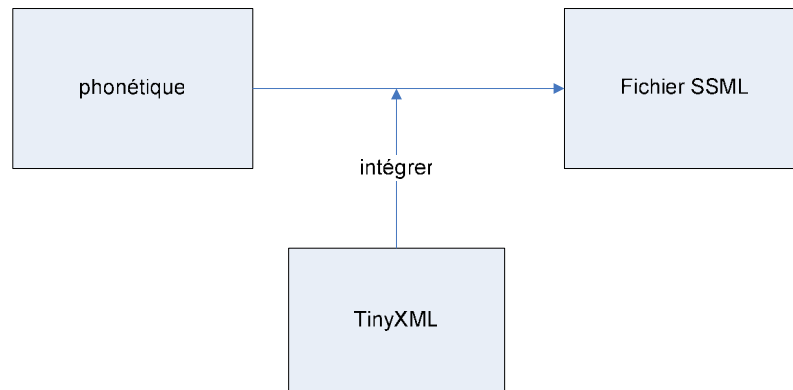


Figure 6. Sortir des résultats dans le fichier SSML à l'aide de TinyXML.

Nous utilisons TinyXML dans 2 cas : lire des fichiers des règles (les fichiers des règles de chaque dialecte et des règles pour prononcer chaque lettre) et sauvegarder les transcriptions phonétiques dans le fichier.

Dans notre travail, des règles sont une partie importante. Nous présentons des règles dans la partie suivante.

2. Règles de transcription phonétique

Pour assurer la flexibilité de notre programme, nous stockons des règles dans les fichiers XML. Nous présentons le format du fichier que nous utilisons. Le fichier des règles standards et celui des règles du Sud sont donnés en exemple aux Annexes A et B.

Une règle simple est représentée comme :

```

<element>
  <écriture type="consonne" position="1">ph</écriture>
  <phoneme>f</phoneme>
</element>
  
```

L'attribut « type » représente le type du token. On a trois types : *consonne* pour les consonnes, *voyelle* pour les voyelles et *semi-voyelle* pour les semi-voyelles.

L'attribut « position » représente la position du token dans la syllabe. La valeur de *position* est un chiffre de 1 à 4 : 1 : premier son, 2 : son intercalé, 3 : son noyau, 4 : dernier son.

Pour les règles qui dépendent du contexte, on ajoute un attribut « follow ». Par exemple : avec une règle : a + {ch} → phonème ě

```

a + {y} → phonème ä
<element>
  <écriture type="voyelle" position="3" follow="ch">a</écriture>
  <phoneme>ě</phoneme>
</element>
  
```

```

<element>
  <ecriture type="voyelle" position="3" follow="y">a</ecriture>
  <phoneme>ă</phoneme>
</element>

```

3. Etapes de fonctionnement du systeme

On réalise notre travail en 3 étapes principales : prétraitement, transcription et sortie en fichier.

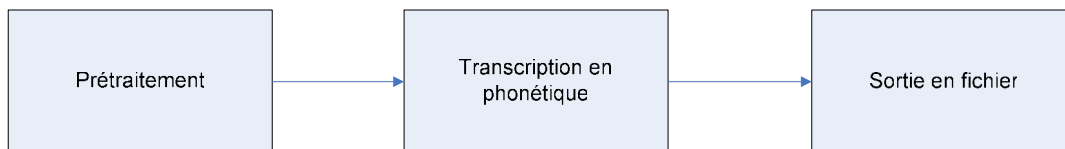


Figure 7. Etapes de transcription phonétique.

Le rôle de chaque étape est présenté dans le Figure 8.

étape	Description
Prétraitement	Découper en tokens et classer les tokens en catégories
Transcription	Transcrire les tokens en phonétique
Sortie en fichier	Stocker le résultat dans le fichier

Figure 8. Description des étapes.

a. Prétraitement :

D'abord, on lit tous des caractères dans le fichier et on stocke dans la mémoire sous forme de chaîne de caractères. Chaque caractère est codé en Unicode 16 bits (suffisant pour le vietnamien et l'API), de type UnicodeChar défini par : *typedef unsigned short UnicodeChar;*

Puis, on fait la segmentation en utilisant des séparateurs les caractères ' ', \t, \r et \n (blanc, tabulation, retour chariot et nouvelle ligne). Chaque token contient alors une chaîne de caractères. Ensuite, on classe des token en trois types suivants :

- Alphabétique où tous les caractères sont des lettres de l'alphabet
- Numérique où tous les caractères sont des chiffres
- Mélange : des lettres, des chiffres et des caractères spéciaux, ce qui n'est pas de l'un des deux types précités.

Avec des tokens de type Mélange, on fait la segmentation de chaque chaîne en sous-chaînes homogènes de l'un des 3 types ci-dessus.

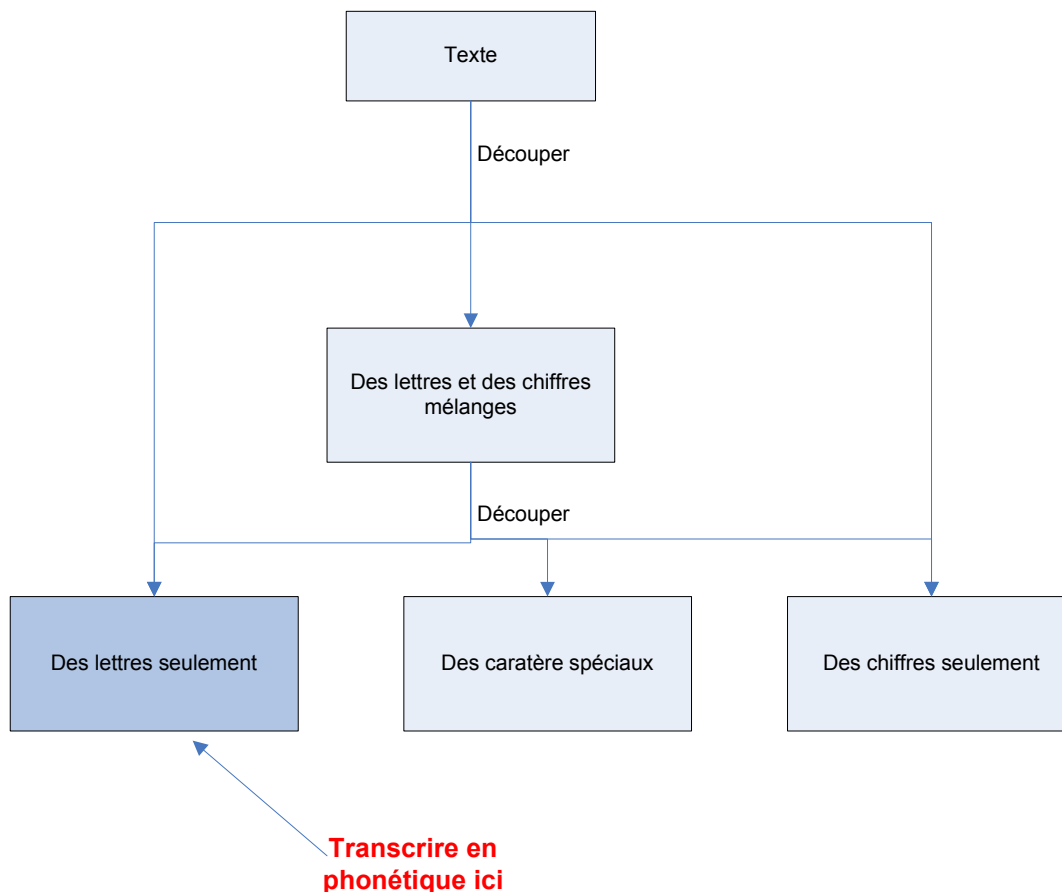


Figure 9. Prétraitement.

On a deux la façons de transcription phonétique. Avec un token, on essaie de transcrire en phonétique selon des règles. Si on obtient des phonèmes, on fait la sortie dans le fichier. Si non, on transcrit lettre par lettre.

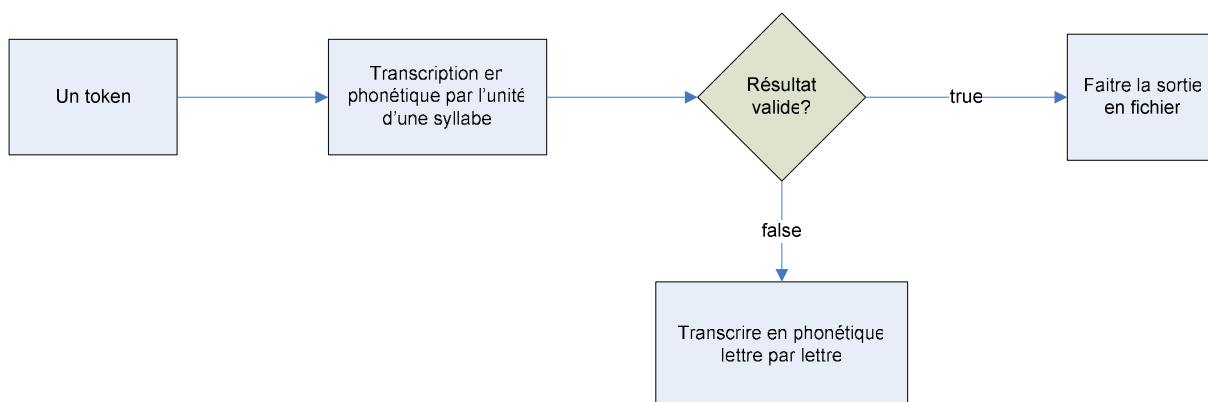


Figure 10. Choisir la methode de transcription phonétique.

b. Transcription phonétique

Comme il existe des mots qui ne sont pas des mots vietnamiens dans des tokens, il faut donc distinguer deux cas différents pour transcrire chaque type de token. Il y a deux options :

- Transcrire en phonétique par l'unité de mot
- Transcrire en phonétique lettre par lettre

❖ Transcrire en phonétique par l'unité de mot

Comme nous avons dit dans la partie Phonétique du vietnamien (Figure 2, à la page 13), comme chaque mot vietnamien se compose 5 parties, on découpe donc chaque token de type mot vietnamien en 5 parties : Ton, premier son, son intercalé, son noyau et dernier son. Le ton vietnamien ne peut pas être bien représenté par l'API, donc on utilise des chiffres (0-5) pour représenter des tons : 0 :sans accent, 1 : accent aigu, 2 : accent grave, 3 :accent retombant, 4 :accent remontant, 5 :accent intensif. Les parties : premier son, son intercalé, son noyau et dernier son sont transcrites en l'API selon des règles disponibles. Alors, chaque mot vietnamien transcrit se compose des caractères de l'API et un chiffre (0-5).

Les règles sont stockées dans des fichiers au format XML et chargées au lancement du programme. Nous présentons aux annexes A et B les fichiers pour des dialectes standard et du Sud.

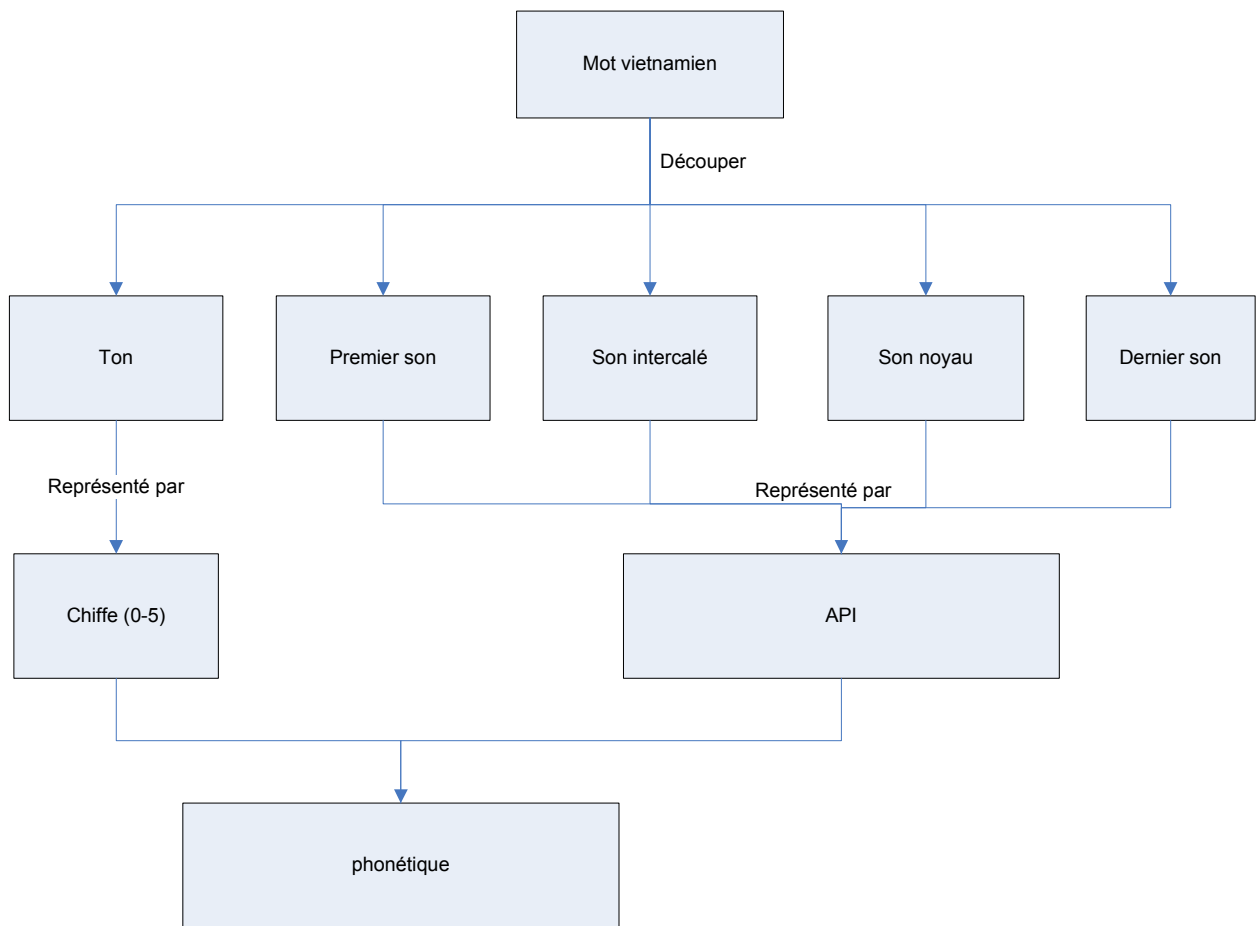


Figure 11. Transcription phonétique par l'unité de mot.

Pour réaliser cela, on a besoin un fichier sous forme XML pour stocker toutes des règles de chaque partie.

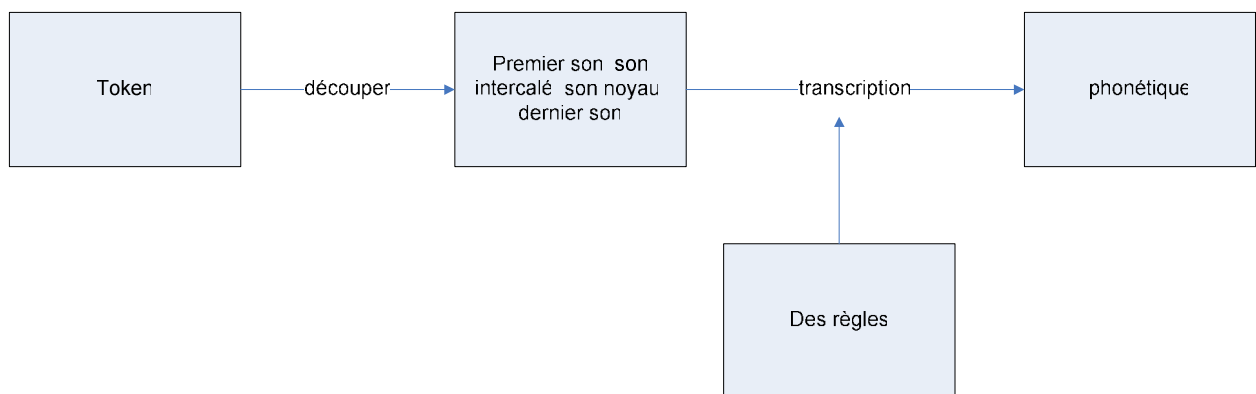


Figure 12. Transcription phonétique d'une syllabe à l'aide des règles.

❖ Transcription phonétique lettre par lettre

Cette opération est plus simple que l'opération de transcription mot par mot. Simplement, on découpe le token en lettres. Pour chaque lettre on transcrit en phonétique.

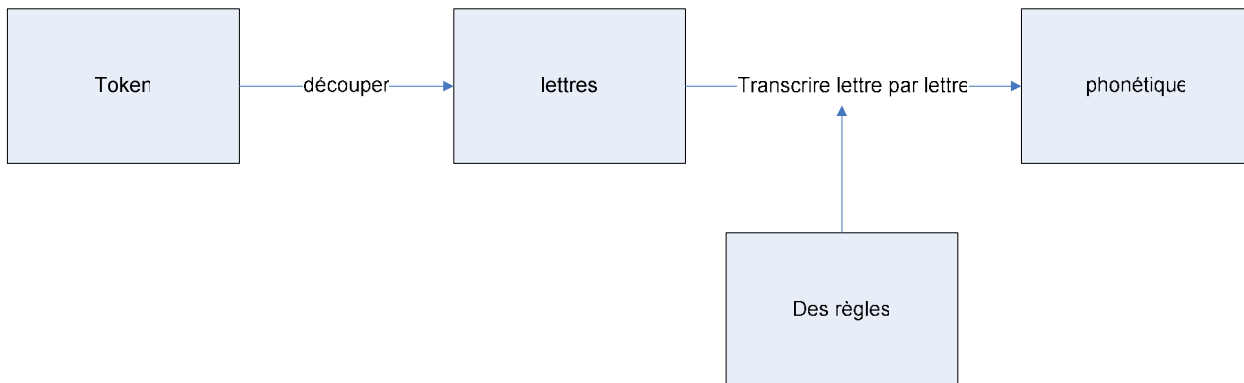


Figure 13. Transcription phonétique lettre par lettre à l'aide des règles.

Les règles de phonétiques utilisées pour la transcription lettre par lettre sont présentées en Annexe C.

c. Sortie dans le fichier :

Il y a deux forme pour le fichier sortie : sous forme texte et sous forme XML. On peut choisir une des deux formes par une option de ligne de commande.

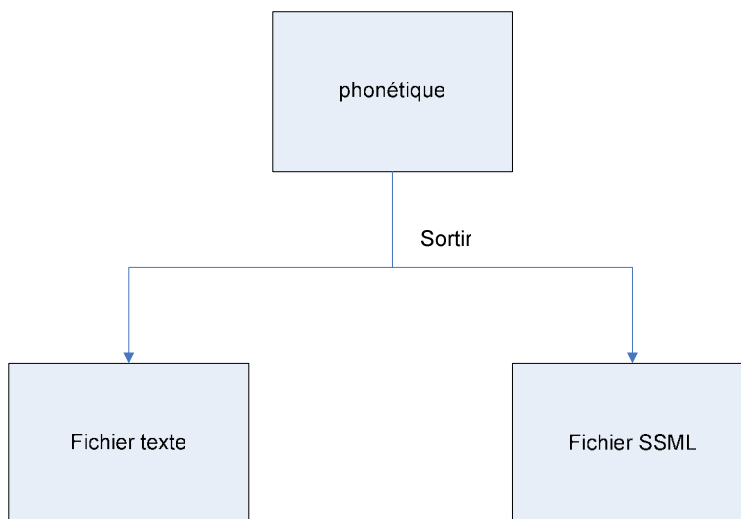


Figure 14. Sortir des résultat dans le fichier.

Le fichier texte est codé en UTF-8. C'est le texte seulement avec des tokens et des phonétiques entre « { » et « } ».

Avec le fichier sous forme XML, on utilise le standard SSML que nous présentons ici en quelques mots. SSML(Speech Synthesis Markup Language) est le langage de balisage de synthèse vocale. Il est développé par le groupe de travail Voice Browser. Ce groupe a tenté de développer des normes permettant un accès au Web au travers d'une interaction verbale. La spécification du langage de balisage de synthèse vocale qui est l'une de ces normes, décrit un langage de balisage riche fondé sur XML pour assister la génération d'une voix synthétisée dans le cadre du Web et dans d'autres applications. Le rôle essentiel du langage de balisage est d'offrir aux créateurs de contenus synthétisables une méthode normalisée pour contrôler les aspects de la voix tels que la prononciation, le volume sonore, la hauteur tonale, le rythme, etc. entre différentes plateformes capables de synthèse vocale. Dans notre projet, on utilise seulement l'élément *phoneme*. L'élément *phoneme* donne une prononciation phonémique/phonétique du texte contenu. L'attribut obligatoire *ph* indique la chaîne de phonème/phone.

L'attribut optionnel *alphabet* désigne l'alphabet phonémique/phonétique. Dans ce contexte, un alphabet se rapporte à un ensemble de symboles représentant les sons d'une ou de plusieurs langues. La valeur valide de cet attribut que nous employons est "ipa" .

Voici un exemple d'un fichier SSML :

```
<?xml version="1.0"?>
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
      http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
      xml:lang="en-US">
  <phoneme alphabet="ipa" ph="t&#x259;mei&#x325;&#x27E;ou&#x325;">
tomato </phoneme>

</speak>
```

VII. Résultats

Le programme est lance depuis la ligne de commande. On a déjà testé dans le système Linux. Voici quelques résultats quand on utilise ce programme.

Commande :

./stp -i test.txt -t testOut.txt

C'est la commande avec le dialecte standard (le programme va utiliser le fichier des règles : standard.xml, c'est le fichier par défaut), le fichier entrée est test.txt, le fichier sortie est testOut.txt. La transcription phonétique est mise entre « { » et « } ».

Le fichier texte résultat est présenté à la figure 15.

```
{4}Đưa {dư_ư0}em {em0}về {ve2}bên {ben0}giác {zǎk1}mơ {mơ0}xanh {sẻ̃n0}
}Khờ {x̣82}đại {zẻ̃i15}thế {t'e1}cỏ {kɔ3}vấn {vẻ̃n4}xanh {sẻ̃n0}màu {mẫu2}cỏ
{kɔ3}Cỏ {kɔ1}nổi {noị4}buồn {bu_on2}lang {lẻ̃n0}thang {t'ẻ̃n0}như {nư0
}ngọn {nɔn5}gió {zɔ1}Cuối {ku_oi1}con {kɔn0}đường {dư_ưn2}mưa {mư_ư0
}trắng {fả̃n1}không {xon0}em {em0}Mưa {mư_ư0}sẻ̃ {sẻ̃4}quên {kn0}làn {lẻ̃n2
}tóc {tɔk1}em {em0}bay {bả̃i0}Đứng {dư_ưn1}ướp {ư_ưp1}mặt {mẻ̃t5}cho {cɔ0
}những {nư_ưn4}lời {ḷi12}gian {zẻ̃n0}dối {zoị1}Nổi {noị4}buồn {bu_on2}không
{xon0}tên {ten0}, {0}nổi {noị4}buồn {bu_on2}mắc {mắk1}tội {toị5}Thanh
{t'ẻ̃n0}ca {kẻ̃0}lâm {lẻ̃m2}thương {t'ư_ưn0}nhớ {ṇ81}nổi {noị4}rêu {zẻ̃u0
}phong {fɔn0}Không {xon0}còn {kɔn2}màu {mẫu2}xanh {sẻ̃n0}chỉ {ci3}cỏ {kɔ1
}minh {mỉn2}anh {ẻ̃n0}Vi {vi0}vút {vut1}bóng {bɔn1}đỏ {do3}dài {zẻ̃i2}loang
{ḷn0}loáng {ḷn1}nước {nư_ưk1}Cầu {kỷu0}thơ {t'ư0}không {xon0}tên {ten0
}câu {kỷu0}thơ {t'ư0}mắc {mắk1}tội {toị5}Cỏ {kɔ1}lẻ̃ {lẻ̃4}là {lẻ̃2}thơ
{t'ư0}viết {vi_ét1}dưới {zư_ưi1}con {kɔn0}mưa {mư_ư0}Em {em0}không
{xon0}về {ve2}phố {fo1}vấn {vẻ̃n4}như {nư0}xưa {sư_ư0}Bàng {bẻ̃n2}đỏ {dɔ3
}lá {lẻ̃1}như {nư0}mùa {mu_o2}xuân {sn0}vấn {vẻ̃n4}thế {t'e1}Viễn {vi_en4
```

Figure 15. Fichier texte du résultat.

Commande :

./stp -i test.txt -x testOut.xml

C'est la commande avec le dialecte standard (le programme va utiliser le fichier des règles : standard.xml, c'est le fichier par défaut), on sort le résultat dans le fichier SSML. Le fichier entrée est test.txt, le fichier de sortie est testOut.xml.

Le fichier SSML résultat est présenté à la figure 16.

```

<?xml version="1.0" ?>
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
      http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
      xml:lang="en-US">
  <phoneme alphabet="ipa" ph="dʉ_ʌ0">Đura</phoneme>
  <phoneme alphabet="ipa" ph="ɛm0">em</phoneme>
  <phoneme alphabet="ipa" ph="vɛ2">vẻ</phoneme>
  <phoneme alphabet="ipa" ph="bɛn0">bên</phoneme>
  <phoneme alphabet="ipa" ph="zʌk1">giác</phoneme>
  <phoneme alphabet="ipa" ph="mɔ0">mơ</phoneme>
  <phoneme alphabet="ipa" ph="sɛŋ0">xanh</phoneme>
  <phoneme alphabet="ipa" ph="xɔ2">Khò</phoneme>
  <phoneme alphabet="ipa" ph="zɛi5">đại</phoneme>
  <phoneme alphabet="ipa" ph="tɛapos;ɛ1">thế</phoneme>
  <phoneme alphabet="ipa" ph="kɔ3">cỏ</phoneme>
  <phoneme alphabet="ipa" ph="vɔm4">vấn</phoneme>
  <phoneme alphabet="ipa" ph="sɛŋ0">xanh</phoneme>
  <phoneme alphabet="ipa" ph="mãɔu2">màu</phoneme>
  <phoneme alphabet="ipa" ph="kɔ3">cỏ</phoneme>
</speak>

```

Figure 16. Fichier SSML du résultat.

Le fichier de sortie est déclaré comme du XML par la ligne :

```
<?xml version="1.0" ?>
```

On déclare la forme de SSML par :

```

<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
      http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
      xml:lang="en-US">

```

Tous les fichiers SSML doivent avoir des deux parties ci-dessus.

Et le reste, ce sont des lignes pour déclarer des tokens et leur transcription phonétique.

Commande :

```
./stp -s -i test.txt -t testOut.txt
```

C'est la commande avec le dialecte du Sud (le programme va utiliser le fichier des règles : sud.xml, c'est le fichier par défaut), on sort le résultat dans le fichier texte. Le fichier d'entrée est test.txt, le fichier de sortie est testOut.txt. On peut voir la différence dans la transcription phonétique entre ce fichier et le fichier précédent avec le dialecte standard.

Đưa {dʷ_ɣ0 }em {ɛm0 }về {je2 }bên {ben0 }giác {jʃk1 }mơ {mʷ0 }xanh {sɛŋ0 }Khò
 {χʌ2 }dại {jɛi5 }thế {t'e1 }cỏ {kɔ3 }vẫn {jʃn4 }xanh {sɛŋ0 }màu {mäu2 }cỏ {kɔ3
 }Cỏ {kɔ1 }nổi {nɣi4 }buồn {bu_on2 }lang {lɛŋ0 }thang {t'ɛŋ0 }như {ŋw0 }ngọn
 {ŋɔn5 }gió {jɔ1 }Cuối {ku_oi1 }con {kɔn0 }đường {dʷŋ2 }mưa {mʷ_ɣ0 }trắng {ʃãŋ1
 }không {χʌŋ0 }em {ɛm0 }Mưa {mʷ_ɣ0 }sẽ {sɛ4 }quên {wen0 }làn {lɛn2 }tóc {tɔk1
 }em {ɛm0 }bay {băi0 }Đứng {dʷŋ1 }rớt {ʋp1 }mặt {mɔt5 }cho {cɔ0 }những {ŋwŋ4
 }lời {lɣi2 }gian {jɛn0 }dối {jɣi1 }Nổi {nɣi4 }buồn {bu_on2 }không {χʌŋ0 }tên
 {tɛn0 }, {0 }nổi {nɣi4 }buồn {bu_on2 }mắc {măk1 }tội {tɣi5 }Thanh {t'ɛŋ0 }ca
 {kɛ0 }làm {lɛm2 }thương {t'ʋŋ0 }nhớ {ŋʌ1 }nổi {nɣi4 }rêu {ʒi_u0 }phong {fɔŋ0
 }Không {χʌŋ0 }còn {kɔn2 }màu {mäu2 }xanh {sɛŋ0 }chỉ {ci3 }cỏ {kɔ1 }mình {miŋ2
 }anh {ɛŋ0 }Vi {ji0 }vút {jʷt1 }bóng {bɔŋ1 }đổ {dɔ3 }dài {jɛi2 }loang {lŋ0 }loáng
 {lŋ1 }nước {nʷk1 }Câu {kɣ_u0 }thơ {t'ʌ0 }không {χʌŋ0 }tên {tɛn0 }câu {kɣ_u0 }thơ
 {t'ʌ0 }mắc {măk1 }tội {tɣi5 }Cỏ {kɔ1 }lẽ {lɛ4 }là {lɛ2 }thơ {t'ʌ0 }viết {ji_et1
 }dưới {jɣi1 }con {kɣn0 }mưa {mʷ_ɣ0 }Em {ɛm0 }không {χʌŋ0 }về {je2 }phổ {fɣ1
 }vẫn {jʃn4 }như {ŋw0 }xưa {sʷ_ɣ0 }Bàng {bɛŋ2 }đổ {dɔ3 }lá {lɛ1 }như {ŋw0 }mùa

Figure 17. Fichier texte du résultat avec le dialecte du Sud.

Le programme a été complé et exécuté sous Linux. Après avoir fini le programmation, nous avons testé notre programme avec plusieurs types du fichier d'entrée : l'article scientifique, le poème, l'article sur internet, quelques morceaux du texte dans le blog, etc.

Puis nous avons testé pour vérifier le changement du dialecte standard et celui du dialecte du Nord.

Après avoir vérifié la transcription phonétique dans les fichiers sortis, la qualité de notre programme est environ 98%. Il reste encore quelques mots qui ne sont pas transcrits très bien en raison des règles contextuelles. C'est la raison de ce problème est l'algorithme pour découper le son intercalé n'est pas très bien car on traite ce problème au niveau de codage, pas des règles. Par exemple, avec les mots avec « u » avant « ê, â » : quên, xuân, on recontre ce problème. Avec des abréviations, on transcrit lettre par lettre, et cette fonction marche très bien, il n'y a pas de problème.

Pour terminer, nous présentons des conclusions et perspectives.

VIII. Conclusion et perspectives

Nous avons réalisé un logiciel de STP indépendant. Ce logiciel peut être réutilisé dans autres logiciels par l'intégration.

Le programme est assez simple et flexible car nous avons stocké des règles dans des fichiers XML. Cela permet de charger facilement le dialecte de chaque région et on peut changer le contenu du fichier sans compiler le programme.

Le programme a supporté des exigences nécessaires du TPE.

Comme nous avons présenté dans la partie **Implémentation**, il y a deux types d'elements lexicaux que nous n'avons pas traite :

- Numérique où tous les caractères sont des chiffres
- Caractères spéciaux.

Pour le type Numérique, il faudra transformer les chiffres sous forme le type Alphabétique, c'est-à-dire on transforme des chiffres en l'écriture en lettres du nombre correspondant.

Avec des caractères spéciaux, on détermine le type de chaque caractère : d'une ponctuation de fin de phrase ('.', ' ?', ' !' ou '...'), d'une ponctuation de milieu de phrase (';', ':', ',' ou '-') ou d'autre chose (mais ne contenant ni lettre, ni chiffre).

Un autre suggestion, avec le type Mélange (qui contient des lettres, des chiffres et des caractères spéciaux, ni de type Alphabétique ni de type Numérique), on détermine si c'est un email, une date, un type d'heure, un nom du site web, une abréviation,etc. pour avoir une façon de transformation sous forme des lettres correspondantes.

Un problème que nous n'avons pas du tant abordé au cours de notre travail est celui de la coarticulation. Très peu d'étude existant sur la coarticulation en vietnamien, il serait donc nécessaire de mener des recherches expérimentales sur ce sujet afin de pouvoir en intégrer les résultats au STP produit.

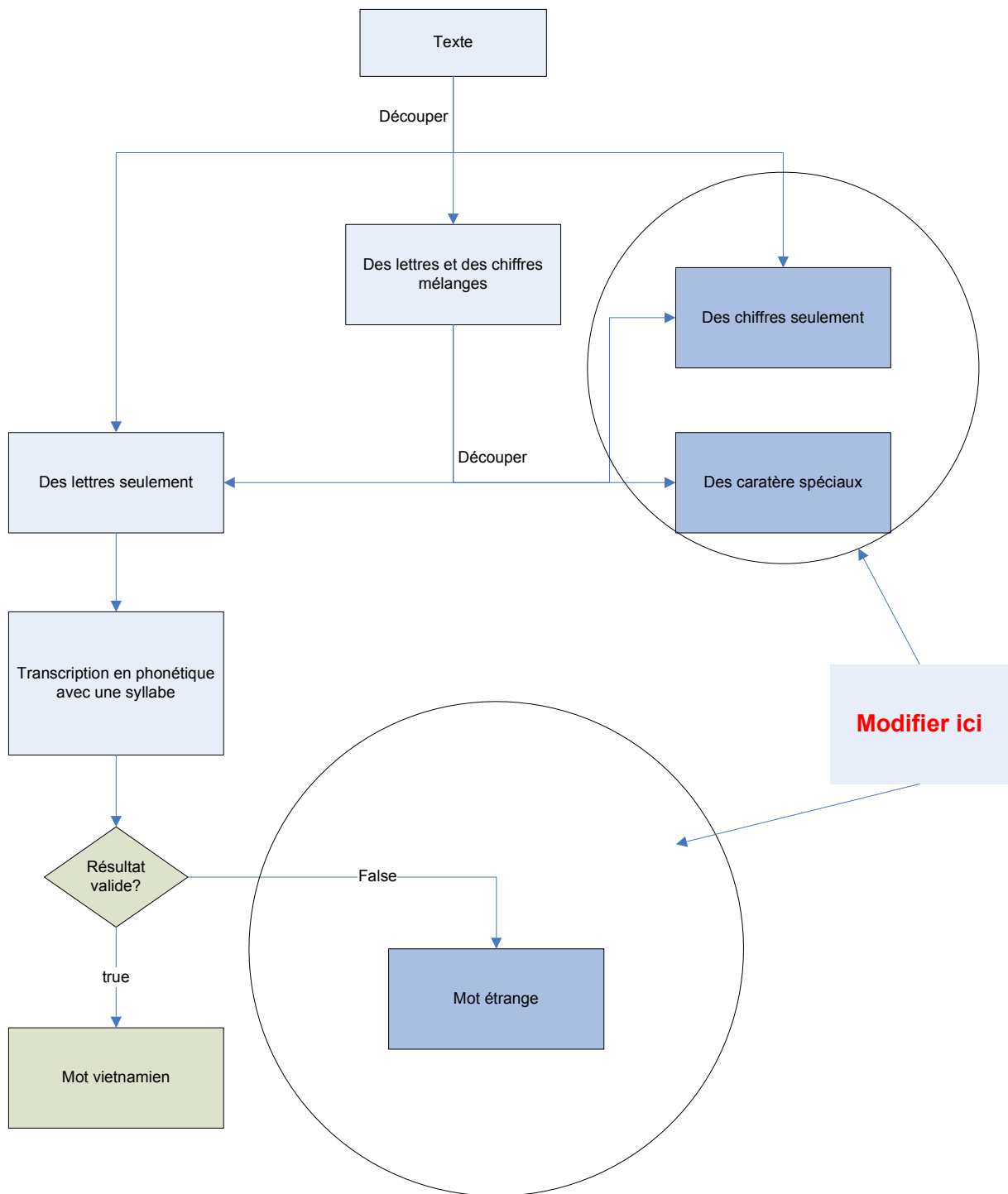


Figure 18. Points d'amélioration possible pour notre travail.

Références

1. Publications

[1]Frederic Bechet, LIA_PHON : un système complet de phonétisation de textes, 2001 Traitement Automatique des Langues - TAL - volume 42 numero 1 - pp 47-67

[2]Hubert Crépy, Christel Amato-Beaujard, Jean-Christophe Marcadet, Claire Waast-Richard, IBM France. *Optimisation d'arbres de décision pour la conversion graphèmes-phonèmes*.XXIVèmes Journées d'Étude sur la Parole, Nancy, 24-27 juin 2002.

[3]NGUYỄN Thị Minh Huyền. *Outils et ressources linguistiques pour l'alignement de textes multilingues français-vietnamiens*. Thèse de doctorat en Informatique, Université Henri Poincaré, Nancy 1, France, 2006.

[4]NGUYỄN Kiên Trường. *Tiếp xúc ngôn ngữ ở Việt Nam*. Nhà Xuất Bản Giáo Dục, 1e édition, 2005.

[5]MAI Ngọc Chừ, VŨ Đức Nghiệu et HOÀNG Ngọc Phiến. *Cơ sở ngôn ngữ học và tiếng Việt*. Nhà Xuất Bản Khoa Học Xã Hội, 7e édition, 2006.

[6]BEAUFORT R., RUELLE A., *eLite : système de synthèse de la parole à orientation linguistique*, Proceedings of JEP 2006, pp. 509-512.

2. Sites web

[7]Lạc Việt (2002), lacviet2002, <http://www.lacviet.com.vn/>

[8]LÊ Hồng Minh (2004), VnSpeech, <http://www.vnisg.net/isg/index.htm>

[9]LÊ Hồng Minh, Quách Tuấn Ngọc. *Một số kết quả phân tích ngữ âm để tổng hợp tiếng Việt từ văn bản bằng luật*, 2005

www.tapchibcv.gov.vn/Uploaded/admin/CS15017_Bai_anh_Minh.pdf

[10]LÊ Tang Hồ (2005), VVV, <http://vietsciences.free.fr/>

[11]LLia Phon. *Guide pour développeur*

<http://download.gna.org/liaphon/devguide.pdf>

[12]MULTITEL. *Synthèse de la parole*

http://www.multitel.be/TTS/Download/plaquettes/synthesis_fr.pdf

[13]Ngôn ngữ học và tiếng Việt : <http://www.ngonngu.net>

[14]Site web de l'API <http://www.arts.gla.ac.uk/ipa/>

[15]Vietnam Lexicography Centre: <http://www.vietlex.com>

Annexe A

Le fichier des règles du dialecte standard

```
<?xml version="1.0"?>
<phonetics>
  <element>
    <ecriture type="consonne" position="1">ph</ecriture>
    <phoneme>f</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">th</ecriture>
    <phoneme>t'</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">tr</ecriture>
    <phoneme>t</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">gi</ecriture>
    <phoneme>z</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">d</ecriture>
    <phoneme>z</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">ch</ecriture>
    <phoneme>c</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="4">ch</ecriture>
    <phoneme>c</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">nh</ecriture>
    <phoneme>ɲ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="4">nh</ecriture>
    <phoneme>ɲ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">ng</ecriture>
    <phoneme>ŋ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="4">ng</ecriture>
    <phoneme>ŋ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">ngh</ecriture>
    <phoneme>ŋ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">kh</ecriture>
    <phoneme>χ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">g</ecriture>
    <phoneme>ɣ</phoneme>
  </element>
</element>
</element>
```

```

                <écriture type="consonne" position="1">gh</écriture>
                <phoneme>ɣ</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">c</écriture>
                <phoneme>k</phoneme>
</element>
<element>
                <écriture type="consonne" position="4">c</écriture>
                <phoneme>k</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">q</écriture>
                <phoneme>k</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">k</écriture>
                <phoneme>k</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">t</écriture>
                <phoneme>t</phoneme>
</element>
<element>
                <écriture type="consonne" position="4">t</écriture>
                <phoneme>t</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">r</écriture>
                <phoneme>ʀ</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">h</écriture>
                <phoneme>h</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">b</écriture>
                <phoneme>b</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">m</écriture>
                <phoneme>m</phoneme>
</element>
<element>
                <écriture type="consonne" position="4">m</écriture>
                <phoneme>m</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">v</écriture>
                <phoneme>v</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">đ</écriture>
                <phoneme>d</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">n</écriture>
                <phoneme>n</phoneme>
</element>
<element>
                <écriture type="consonne" position="4">n</écriture>
                <phoneme>n</phoneme>
</element>
<element>
                <écriture type="consonne" position="1">l</écriture>
                <phoneme>l</phoneme>

```

```

</element>
<element>
  <écriture type="consonne" position="1">x</écriture>
  <phoneme>s</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">p</écriture>
  <phoneme>p</phoneme>
</element>
<element>
  <écriture type="consonne" position="4">p</écriture>
  <phoneme>p</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">s</écriture>
  <phoneme>ξ</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">i</écriture>
  <phoneme>i</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">y</écriture>
  <phoneme>i</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ê</écriture>
  <phoneme>e</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">e</écriture>
  <phoneme>ε</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3" follow="nh">a</écriture>
  <phoneme>ě</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3" follow="ch">a</écriture>
  <phoneme>ě</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">a</écriture>
  <phoneme>a</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ă</écriture>
  <phoneme>ă</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3" follow="y">a</écriture>
  <phoneme>ă</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3" follow="u">a</écriture>
  <phoneme>ă</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">σ</écriture>
  <phoneme>ϣ</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">â</écriture>
  <phoneme>ř</phoneme>
</element>

```

```

<element>
  <écriture type="voyelle" position="3">u</écriture>
  <phoneme>u</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ô</écriture>
  <phoneme>o</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3" follow="">o</écriture>
  <phoneme>ɔ</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">o</écriture>
  <phoneme></phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">oo</écriture>
  <phoneme>ɔ</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">u</écriture>
  <phoneme>u</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ia</écriture>
  <phoneme>i_e</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ya</écriture>
  <phoneme>i_e</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">iê</écriture>
  <phoneme>i_e</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">yê</écriture>
  <phoneme>i_e</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">uô</écriture>
  <phoneme>u_o</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ua</écriture>
  <phoneme>u_o</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">uɔ</écriture>
  <phoneme>u_x</phoneme>
</element>
<element>
  <écriture type="voyelle" position="3">ura</écriture>
  <phoneme>u_x</phoneme>
</element>
<element>
  <écriture type="semi-voyelle" position="4">i</écriture>
  <phoneme>j</phoneme>
</element>
<element>
  <écriture type="semi-voyelle" position="4">y</écriture>

```

```
        <phoneme>j</phoneme>
</element>
<element>
  <écriture type="semi-voyelle" position="2">o</écriture>
  <phoneme>ɥ</phoneme>
</element>
<element>
  <écriture type="semi-voyelle" position="4">o</écriture>
  <phoneme>ɥ</phoneme>
</element>
<element>
  <écriture type="semi-voyelle" position="2">u</écriture>
  <phoneme>ɥ</phoneme>
</element>
<element>
  <écriture type="semi-voyelle" position="4">u</écriture>
  <phoneme>ɥ</phoneme>
</element>
</phonetics>
```

Annexe B

Le fichier des règles du dialecte du Sud

```
<?xml version="1.0"?>
<phonetics>
  <element>
    <ecriture type="consonne" position="1">ph</ecriture>
    <phoneme>f</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">th</ecriture>
    <phoneme>t</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">tr</ecriture>
    <phoneme>t</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">gi</ecriture>
    <phoneme>j</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">d</ecriture>
    <phoneme>j</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">ch</ecriture>
    <phoneme>c</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="4">ch</ecriture>
    <phoneme>c</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">nh</ecriture>
    <phoneme>ɲ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="4">nh</ecriture>
    <phoneme>ɲ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">ng</ecriture>
    <phoneme>ŋ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="4">ng</ecriture>
    <phoneme>ŋ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">ngh</ecriture>
    <phoneme>ŋ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">kh</ecriture>
    <phoneme>χ</phoneme>
  </element>
  <element>
    <ecriture type="consonne" position="1">g</ecriture>
    <phoneme>ɣ</phoneme>
  </element>

```

```

</element>
<element>
  <écriture type="consonne" position="1">gh</écriture>
  <phoneme>ɣ</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">c</écriture>
  <phoneme>k</phoneme>
</element>
<element>
  <écriture type="consonne" position="4">c</écriture>
  <phoneme>k</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">q</écriture>
  <phoneme>w</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">k</écriture>
  <phoneme>k</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">t</écriture>
  <phoneme>t</phoneme>
</element>
<element>
  <écriture type="consonne" position="4">t</écriture>
  <phoneme>t</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">r</écriture>
  <phoneme>ʒ</phoneme>
</element>
<element>
  <écriture type="consonne" position="1" follow="o">h</écriture>
  <phoneme>w</phoneme>
</element>
<element>
  <écriture type="consonne" position="1" follow="u">h</écriture>
  <phoneme>w</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">b</écriture>
  <phoneme>b</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">m</écriture>
  <phoneme>m</phoneme>
</element>
<element>
  <écriture type="consonne" position="4">m</écriture>
  <phoneme>m</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">v</écriture>
  <phoneme>j</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">đ</écriture>
  <phoneme>d</phoneme>
</element>
<element>
  <écriture type="consonne" position="1">n</écriture>
  <phoneme>n</phoneme>
</element>
<element>

```

```

                <ecriture type="consonne" position="4">n</ecriture>
                <phoneme>n</phoneme>
</element>
<element>
                <ecriture type="consonne" position="1">l</ecriture>
                <phoneme>l</phoneme>
</element>
<element>
                <ecriture type="consonne" position="1">x</ecriture>
                <phoneme>s</phoneme>
</element>
<element>
                <ecriture type="consonne" position="4">p</ecriture>
                <phoneme>p</phoneme>
</element>
<element>
                <ecriture type="consonne" position="1">s</ecriture>
                <phoneme>s</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3">i</ecriture>
                <phoneme>i</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3">y</ecriture>
                <phoneme>i</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="u">ê</ecriture>
                <phoneme>i</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="m">ê</ecriture>
                <phoneme>i</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="p">ê</ecriture>
                <phoneme>i</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="n">ê</ecriture>
                <phoneme>a</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="t">ê</ecriture>
                <phoneme>a</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="c">ê</ecriture>
                <phoneme>a</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3"follow="">ê</ecriture>
                <phoneme>e</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3">e</ecriture>
                <phoneme>ε</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3" follow="nh">a</ecriture>
                <phoneme>ǣ</phoneme>
</element>
<element>
                <ecriture type="voyelle" position="3" follow="ch">a</ecriture>

```

```

        <phoneme>ě</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">a</écriture>
    <phoneme>a</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">ä</écriture>
    <phoneme>ä</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3" follow="y">a</écriture>
    <phoneme>ä</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3" follow="u">a</écriture>
    <phoneme>ä</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">σ</écriture>
    <phoneme>ϣ</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">â</écriture>
    <phoneme>ǎ</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">u</écriture>
    <phoneme>u</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">ô</écriture>
    <phoneme>ϣ</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3" follow="">o</écriture>
    <phoneme>ɔ</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">o</écriture>
    <phoneme></phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">oo</écriture>
    <phoneme>ɔ</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">u</écriture>
    <phoneme>u</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">ia</écriture>
    <phoneme>i_e</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">ya</écriture>
    <phoneme>i_e</phoneme>
</element>
<element>
    <écriture type="voyelle" position="3">iê</écriture>
    <phoneme>i_e</phoneme>
</element>
<element>

```

```

        <ecriture type="voyelle" position="3">yê</ecriture>
        <phoneme>j_e</phoneme>
    </element>
    <element>
        <ecriture type="voyelle" position="3">uô</ecriture>
        <phoneme>u_o</phoneme>
    </element>
    <element>
        <ecriture type="voyelle" position="3">ua</ecriture>
        <phoneme>u_o</phoneme>
    </element>
    <element>
        <ecriture type="voyelle" position="3">uɔ</ecriture>
        <phoneme>u</phoneme>
    </element>
    <element>
        <ecriture type="voyelle" position="3">ura</ecriture>
        <phoneme>u_x</phoneme>
    </element>
    <element>
        <ecriture type="semi-voyelle" position="4">i</ecriture>
        <phoneme>j</phoneme>
    </element>
    <element>
        <ecriture type="semi-voyelle" position="4">y</ecriture>
        <phoneme>j</phoneme>
    </element>
    <element>
        <ecriture type="semi-voyelle" position="2">o</ecriture>
        <phoneme></phoneme>
    </element>
    <element>
        <ecriture type="semi-voyelle" position="4">o</ecriture>
        <phoneme>ɥ</phoneme>
    </element>
    <element>
        <ecriture type="semi-voyelle" position="2">u</ecriture>
        <phoneme></phoneme>
    </element>
    <element>
        <ecriture type="semi-voyelle" position="4">u</ecriture>
        <phoneme>ɥ</phoneme>
    </element>
</phonetics>

```

Annexe C

Le fichier des règles pour la transcription lettre par lettre

```
<?xml version="1.0"?>
<phonetics>
  <letter writing = "d">de</letter>
  <letter writing = "g">gɣ2</letter>
  <letter writing = "c">xe</letter>
  <letter writing = "q">kɣj</letter>
  <letter writing = "k">ka</letter>
  <letter writing = "t">te</letter>
  <letter writing = "r">ε rɣ2</letter>
  <letter writing = "h">hat1</letter>
  <letter writing = "b">be</letter>
  <letter writing = "m">εm mɣ2</letter>
  <letter writing = "v">ve</letter>
  <letter writing = "d̃">de</letter>
  <letter writing = "n">εn nɣ2</letter>
  <letter writing = "l">ε lɣ2</letter>
  <letter writing = "x">ic1 si2</letter>
  <letter writing = "p">pe</letter>
  <letter writing = "s">εt1 si2</letter>
  <letter writing = "i">i</letter>
  <letter writing = "y">i</letter>
  <letter writing = "ê">e</letter>
  <letter writing = "e">ε</letter>
  <letter writing = "a">a</letter>
  <letter writing = "ä">ä</letter>
  <letter writing = "o">ɣ</letter>
  <letter writing = "â">ɣ</letter>
  <letter writing = "u">u</letter>
  <letter writing = "u">u</letter>
  <letter writing = "ô">o</letter>
  <letter writing = "o">ɔ</letter>
  <letter writing = "f">ε fɣ2</letter>
  <letter writing = "w">ve kεp1</letter>
</phonetics>
```