**INSTITUT POLYTECHNIQUE DE HANOI**

**PROGRAMME DE FORMATION D'INGÉNIEURS D'EXCELLENCE AU VIETNAM (PFIEV)**

**---------------\*---------------**



**PROJET DE FIN D'ETUDES**

**SUJET:**

**INTERACTION AVEC LES TELEPHONES INTELLIGENTS UTILISANT LA VOIX**

Etudiant : NGUYỄN DUY BÌNH

Filière : Informatique Industrielle - PFIEV C53

Tuteur  : Dr.TRẦN ĐỖ ĐẠT

HANOI 6/2013

[CHAPITRE 1. INTRODUCTION 3](#_Toc359236642)

[1.1 Problème. 3](#_Toc359236643)

[1.2 Projet VIVA 3](#_Toc359236644)

[1.2.1 Introduction 3](#_Toc359236645)

[1.2.2 Fonction 3](#_Toc359236646)

[1.2.3 Mes fonctions 3](#_Toc359236647)

[CHAPITRE 2. BASE THÉORIQUE ET ARCHITECTURE DU SYSTÈME VIVA 4](#_Toc359236648)

[2.1 Schéma de bloc du système 4](#_Toc359236649)

[2.1.1 Schéma de bloc 4](#_Toc359236650)

[2.1.2 Description des blocs 5](#_Toc359236651)

[2.2. L’architecture VIVA sur l’Android 6](#_Toc359236652)

[CHAPITRE 3. BLOCS DE FONCTION DE MESSAGES 9](#_Toc359236653)

[3.1 Envoyer un message. 9](#_Toc359236654)

[3.2 Lire le message entrant. 10](#_Toc359236655)

[CHAPITRE 4. AJOUTER LES SIGNES DIACRITIQUES AUX TEXTES EN VIETNAMIEN SANS SIGNE 11](#_Toc359236656)

[4.1. Les raisons pour résoudre le problème 11](#_Toc359236657)

[4.2. Situation actuelle du problème. 11](#_Toc359236658)

[4.2.1. Les approches actuelles 11](#_Toc359236659)

[4.2.3. Approche en utilisant la traduction automatique (traduction automatique). 12](#_Toc359236660)

[4.3. La traduction automatique Moses. 12](#_Toc359236661)

[4.3.1. L’introduction de la traduction automatique Moses. 12](#_Toc359236662)

[4.3.2. Fondements théoriques de la traduction automatique statistique. 12](#_Toc359236663)

[4.4. Utiliser Moses pour le probème d’ajouter les signes Vietnamiens 13](#_Toc359236664)

[4.4.1. Étape 1: Installer Moses Toolkit. 14](#_Toc359236665)

[4.4.2. Étape 2: Créer corpus. 14](#_Toc359236666)

[4.4.3. Étape 3: Entraîner le modèle de langue. 14](#_Toc359236667)

[4.4.4. Étape 4: Entraîner le système 15](#_Toc359236668)

[4.4.5. Étape 5: Contrôler: 15](#_Toc359236669)

[4.5. Déploiement le module d'ajouter les signes au texte pour VIVA. 15](#_Toc359236670)

[4.5.1. Créer le serveur de traduction 15](#_Toc359236671)

[4.5.2. Construire le module d’ajouter dans VIVA 15](#_Toc359236672)

[4.6 Estimation du système 15](#_Toc359236673)

[4.6.1 Estimation par la comparaison Word-By-Word. 15](#_Toc359236674)

[CHAPITRE 5. PROGRAMME VIVA DE VERSION DEMO 16](#_Toc359236675)

[5.1 Joindre des modules 16](#_Toc359236676)

[5.2 Demo du programme VIVA 16](#_Toc359236677)

[CHAPITRE 6. RÉSUMÉ ET ÉVALUATION 16](#_Toc359236678)

[6.1 Conclusion générale 16](#_Toc359236679)

[6.2 Orientation de développement 16](#_Toc359236680)

# INTRODUCTION

## 1.1 Problème.

Les appareils portables intelligents tels que Smartphone, Tablet se développe à un rythme rapide et deviennent de plus en plus une partie intégrante de la vie de l'homme moderne. Communiquer avec les smartphone par la voix est un sujet intéressant, une des tendances technologiques actuelles, avec l'objectif de transformer le portable en un secrétaire virtuelle pour les utilisateurs, d’aider les utilisateurs à effectuer des tâches sans du besoin de toucher l'écran, créant ainsi une expérience pratique et agréable.

De nombreux fabricants smartphones, sociétés de technologie logicielle en premier rang du monde investissent, font la recherche et le développement des systèmes interactifs avec les smartphones par la voix comme Apple, Google, Nuance, Samsung ... Produits Siri, Google Now, Nuance Talks, S Voice ...

Cependant, il n'a jamais été un système qui permet aux utilisateurs d'interagir avec le portable en vietnamien, donc nous avons choisi ce sujet comme thèse de mémoire de fin d'études.

## 1.2 Projet VIVA

### 1.2.1 Introduction

Pour mettre en œuvre le projet en interagissant avec TiengViet smartphone, projet VIVA (Voice adjoint vietnamien) a été créé à l’institut de recherche MICA dans le but :

- Audible: la reconnaissance vocale (vietnamien).

- Compréhensible: comprendre le commandement de l'utilisateur.

- Dire: commentaires des utilisateurs par la voix de nouveau.

- Faire secrétaire permet à l'utilisateur d'effectuer des actions sur votre téléphone sans avoir à toucher le clavier pour contrôler, l'interaction sera été demandé, la notification des résultats et des informations par la voix.

### 1.2.2 Fonction

Dans le cadre du projet de fin, et le groupe a pour objectif de compléter le bloc de reconnaissance de la parole, bloc de compréhension de commande, bloc de synthèse, conception et construction d'infrastructures pour les applications VIVA basées sur Android 2.3 avec les fonctions de base: envoyer un message, lire le message, répondre la météo, le prix de l'or, le prix.

### 1.2.3 Mes fonctions

* Concevoir l'architecture du programme VIVA.
* Mettre en œuvre le module de fonction message.
* Restaurer signes diacritiques pour le message contenu signes diacritiques manquantes.

# BASE THÉORIQUE ET ARCHITECTURE DU SYSTÈME VIVA

## 2.1 Schéma de bloc du système

### 2.1.1 Schéma de bloc

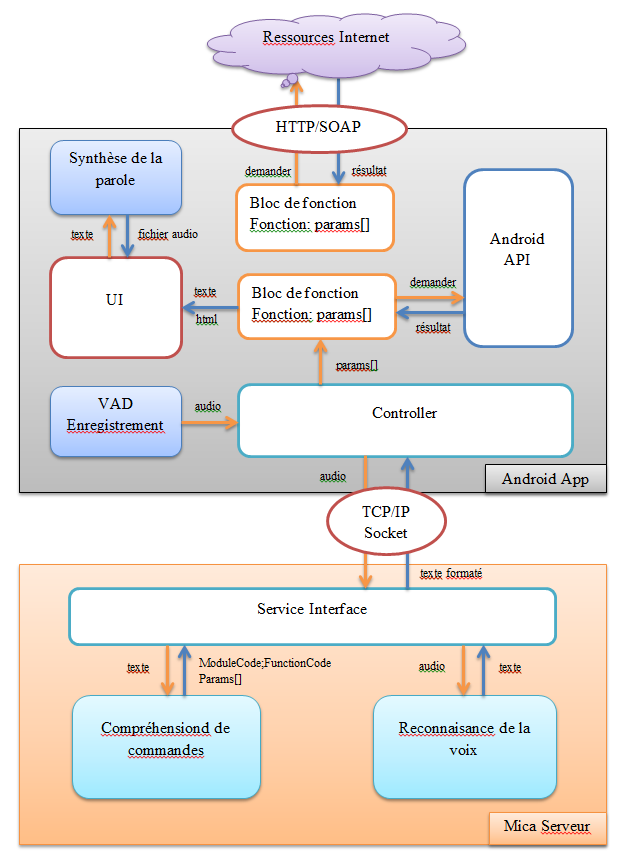
****

Figure 2.1 Schéma de bloc du système VIVA

### 2.1.2 Description des blocs

Le système VIVA comporte les blocs principaux suivants:

1. ***Bloc VAD (Voice Automatic Detection) :***

* **Description :** La voix du utilisateur est enregistrée, est passée le bloc VAD qui détecte automatiquement la voix pour commencer à enregistrer, arrêter automatiquement d’enregistrer quand l’utilisateur arrête de parler aussi un intervalle prédéfini ( 0.5 secondes ). Après la réduction du bruit et la compression de fichier, le bloc VAD envoye le fichier d’enregistrement au bloc de contrôle pour envoyer au serveur de reconnaissance.
* **Entrée :** de la voix
* **Sortie :** fichier d’enregistrement

1. ***Bloc Controller ( bloc de commande ):***

* **Description:** Bloc Controller est de communiquer avec le Serveur qui exécute le bloc de reconnaissance et de compréhension par socket TCP / IP. Bloc Controller envoie des fichiers vocaux qui ont été collectées et traitées par le bloc VAD au Serveur après avoir obtenu les résultats formatés texte contenant le contenu bulletin d'information du système de reconnaissance, le code de module de fonction, le code de commande et la valeur du paramètre seront effectués. A partir du résultat retourné, le bloc Controller cherche un module de fonction et d'appeler le correspondant commande avec les paramètres d'entrée dans le cas échéant.
* **Entrée:** Fichier enregistrement du bloc VAD.
* **Sortie:** Activer le module de fonction et les commandes correspondantes.

1. ***Bloc Service Interface (Bloc d’interface de services giao installé au Serveur MICA):***

* **Description:** Le bloc doivent recevoir des demandes socket TCP / IP des client qui sont envoyés par le bloc Controller. A partir du fichier enregistrés envoyés, ce bloc appelle au bloc de reconnaissance pour obtenir la phrase correspondant en text, après il appelle au bloc de compréhension pour analyser la demande de l’utilisateur et renvoyer les résultats au client.
* **Entrée:** Fichier d’enregistrement
* **Sortie:** Segment de text qui place la phrase reconnu et le résultat analysé par le bloc de compréhension

1. ***Bloc de reconnaisance de la voix:***

* **Description:** Le bloc fait des tâches de reconnaître les fichiers enregistrés et de renvoyer les parols corresspondantes en text
* **Entrée:** Fichier enregistré.
* **Sortie:** Text.

1. ***Bloc de compréhensiond de commandes***

* **Description:** Le bloc fait le tâche d'analyser la sémantique des commandes de l'utilisateur pour déterminer si le système doit appeler quel module de fonction qui va exécuter la commande avec les paramètres de l'instruction extrait dans la commande pour la réponse à l'utilisateur.Par exemple, si l'utilisateur demande que"Je veux voir des informations météorologiques à Hanoi demain", ce bloc va analyser et renvoyer comme :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *ModuleCode* | QUERYINFO | *FunctionCode* | WEATHER | *Param* | Hà nội | *Param* | Ngày mai |

* **Entrée:** Text.
* **Sortie:** Text formaté

1. ***Blocs de fonction:***

* **Description:** Les blocs fonctionnels comportent les « functions » dont la mise en œuvre de la fonction est d’exécuter de tâche requise par l'utilisateur, ce bloc a été construit dans la liste des fonctions du VIVA énumérés dans la section 1.3.3. Les fonctions après sont activées par le bloc de Contrôleur sont exécutés un scénario pré-programmée, le processus d'exécution peut être renvoyé à l'API du système d'exploitation pour exécuter des tâches (SMS, appels téléphoniques, e-mail ...) ou pour interroger les sources d'information sur Internet afin d’obtenir d’informations pour la réponse à l’utilisateur (météo, taux de change, l'or, les stocks ...). Une fois exécuté avec succès, il va appeler àl'interface d’utilisateur ( bloc UI ) pour afficher les résultats sous forme de texte ou HTML.
* **Entrée:** Tham số.
* **Sortie:** Text, Html result.

1. ***Bloc UI ( Interface d’utilisateur ):***

* **Description:** Ce bloc fait le tâche d’affichier les résultats en texte - la réponse de VIVA et des données concrets en HTML (prix de l'or, météo ...) aux utilisateurs. Avec les réponses affichées sur l'écran de VIVA, le bloc UI va appeler au bloc de synthèse de la parole pour obtenir des fichiers vocaux et émetre la réponse aux utilisateurs.
* **Entrée:** La réponse en text, donnée affiché en Html
* **Sortie:** Résultats sur écran

1. ***Bloc de synthèse de la parole:***

* **Description:** Ce bloc fait le tâche de synthétiser la phrase entrée en text et renvoyer le fichier vocal en Khối làm nhiệm vụ tổng hợp câu nói dạng text đưa vào và trả về file vocal sonore.
* **Entrée:** Text.
* **Sortie:** Fichier vocal
* Bloc de reconnaissance de la parole et de la compréhension sémantique ont des structures complexes, en exécutant ils s’occupent des ressources des forces de l'occupation (espace disque, mémoire, CPU) beaucoup, donc ne devrait pas encore être très grand intégré dans des applications sur le téléphone dialogue qui est déployée sur le serveur de MICA, pour le client de se connecter en utilisant par socket TCP / IP. Pour cette raison, le système VIVA en fonctionnant a toujours besoin de connexion d’Internet.

## 2.2. L’architecture VIVA sur l’Android

L’application VIVA est conçu et construit sur Android-un système d’exploitation de source ouvert puissant pour les dispositifs mobiles. OS Android est de plus en plus forte et le nombre d'appareils qui sont installé ce système d'exploitation également en constante augmentation. Programme VIVA est construit sur l’étage d’application.

VIVA est analysé et conçu orienté d’objet. L’architecture VIVA comporte les packages:

* *com.mica.viva:* package du programme
* *com.mica.viva.controller*: principal bloc de demande
* *com.mica.via.controller.sms*: bloc de fonction de message
* *com.mica.viva.controller.queryinformation*:bloc de fonction de recherche d'information
* *com.mica.viva.inputting*: bloc d’enregistrement, VAD
* *com.mica.viva.entity*: bloc d’entité, contient les classe d'entité qui contient les information des modules de fonction et des paramètres de fonction ...
* *com.mica.viva.resource*: lit des fichiers de donnée, gére les entités
* *com.mica.viva.utility* : contient des classes utilitaires.

Diagramme de classes du programme:



Figure 2.2 Diagramme de classes du programme VIVA

# BLOCS DE FONCTION DE MESSAGES

## 3.1 Envoyer un message.



Figure 3.1 Le diagramme d'activités de fonctionnalités de messagerie

## 3.2 Lire le message entrant.



Figure 3.2 Le diagramme d'activité d'une fonction de lire le message entrant

# AJOUTER LES SIGNES DIACRITIQUES AUX TEXTES EN VIETNAMIEN SANS SIGNE

## 4.1. Les raisons pour résoudre le problème

Dans la fonction de dire message reçu du VIVA, il y a un problème, c’est que le contenu du message est normalement écrit en forme sans signe par l’utilisateur pour pour accélérer la rédaction ainsi que le contenu des nouvelles. Le contenu en vietnamient sans signe n’est pas synthétisé en entrant le bloc de synthèse ou la sortie difficile de comprendre, aberrant, donc il est neccaissaire de normaliser, d’ajouter automatiquement des signes pour le contenu avant d’entrer dans le bloc de synthèse.

## 4.2. Situation actuelle du problème.

Problème pour ajouter les signes à un texte est posé assez tôt avec de nombreuses langues dans le monde. Seulement en Europe, il y a jusqu'à 36 langues en utilisant les lettres accentuées lating comme le vietnamien.

Ajouter automatiquement des signes pour le texte vietnamien est vraiment un travail n'est pas facile, la plus grande difficulté, c'est qu'il ya beaucoup de choix quand se déplaçant de mots sans signes à mots de signe avec des significations différentes qui rend l'importation confusion. Un mot est ajouté les signes exactement dans le contexte des relations sémantiques spécifiques avec d'autres mots dans la phrase. Par conséquent problème d’ajouter les signes vietnamiens a été fixé il y a longtemps, mais les recherches sur ce problème n'a pas vraiment gagné beaucoup de succès

Table 4.1 La proportion de mots de signe et la proportion ambigue de mots sans signes dans quelques langues

| **Langue** | **Proportion de mots de signe** | **Proportion ambigue de mots sans signes** |
| --- | --- | --- |
| French [4] | 15% | 50% |
| Romanian [2] | 35% | 25% |
| Vietnamese[5] | 95% | 80% |

### 4.2.1. Les approches actuelles

Les deux principales approches: l'approche basée sur le mot (mot-based) et les approches fondées sur les syllabes (basée sur des caractères).[3]

* L'approche est basée sur le mot: activité basée sur l'analyse du vocabulaire et de modèle de langage, le procédé selon cette approche nécessite un dictionnaire complet, les grands corpus avec les étapes d’analyse de simples vocabulaire comme le clivage, l'analyse de syntaxte, étiqueté (le type, la syntaxe de l'étiquette, étiquette sémantique).
* Avantages: grande précision.
* Défauts:
  + Car l’opération basé sur le modèle de language donc ces méthodes dépendront de chaque langue différente
  + Il faut beaucoup d'effort pour construire le dictionnaire de données, corpus, outre les données de corpus par rapport à la norme de grammaire, sémantique.
  + Il faut avoir beaucoup de temps de traitement afin d'effectuer des tâches telles que le clivage, l'analyse, le marquage ...
* L'approche est basée sur les syllabes: les méthodes fondées sur cette approche fonctionnent avec les différents algorithmes, basés sur des informations statistiques à partir de n-grammes extraites des données d'entraînement. Les données d'entraînement juste à l'état brut, pas nécessairement à travers les étapes de traitement telles que le clivage, étiqueté.
  + Avantages: Ne comptez pas sur la langue, le déploiement facile, processeur d’haute vitesse.
  + Inconvénients:

Avec le langage dont le tâche d’ajouter des signes dépend du contexte et sujettes à ambiguïté comme le vietnamien, le taux d'exactitude de la méthode est assez faible et généralement appliquée au langage simple (peu de signe, à l'aussi simple que cela de son, ...)

### 4.2.3. Approche en utilisant la traduction automatique (traduction automatique).

L'idée de base de cette méthode est: le vietnamien considéré comme non signé (ou l'absence de signes) est une nouvelle langue et utilise la traduction automatique pour traduire du texte à partir de ce nouveau langage aux normes vietnamiennes. Construction d'un système de traduction automatique est un qui n'est pas facile, pour mettre en œuvre cette méthode, j’ai utilisé un système de traduction automatique de source ouvert nommé Moses.

## 4.3. La traduction automatique Moses.

### 4.3.1. L’introduction de la traduction automatique Moses.

Moses est un système de traduction automatique (Machine Translation - MT) en utilisant des méthodes statistiques, les systèmes de traduction automatique statistique sont formés par le service de grandes quantités de données en parallèle (à partir de ces données, le système apprend à traduire les petits segments.[6]

### 4.3.2. Fondements théoriques de la traduction automatique statistique.

Traduction automatique statistique est entendu que traduire le texte d'une langue vers une autre langue basée sur le modèle généré automatiquement à partir d'un corpus.[1]

Avec l'exemple de la phrase anglaise a l’entrée e, trouver la phrase vietnamienne V tel que la probabilité conditionnelle P (v|e) est le plus grand. Depuis v et e sont interdépendants, donc en théorie la probabilité conditionnelle formule de Bayes:

argmax(P(v|e)) = argmax(P(v) \* P (e|v))

Parmi eux:

* P (v) est le modèle de langage
* P (e|v) est le modèle de traduction

Décodeur (decoder)

argmax(P(v|e)) = argmax(P(v) \* P (e|v))

Modèle de langue P(v)

Modèle de traduction P(e|v)

Corpus monolingue

Corpus bilingue

Phrase originale e: ‘I am student’

Phrase destinataire v: ‘tôi là sinh viên’

Figure 4.1Modèle du système de la traduction automatique statistique

* **Modèle de traduction**

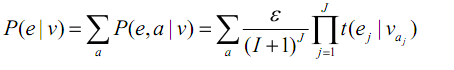
Calculer la probabilité P (e|v) basée sur l'alignement de mot (Word-alignement) entre l'anglais (langue source) - le vietnam (la langue cible).

Pour la paire (v, e), il existe de nombreux liens de différentes manières, afin d'évaluer la précision de chaque lien doit être intéressés par le lien probabilité P (A = | V = v , E = e) est constitué de trois variables aléatoires (V, A, E):

, trong đó

* *Le modèle est basé sur des unités de mot*

Proposé par IBM, l'hypothèse de choisir simplement d'aligner sur des unités de mot. La probabilité P (e|v) est calculé:



L'inconvénient de ce modèle est résolu seulement aligner 1-1, 1-plusieurs, ne peut pas résoudre l’alignement de plusieurs - plusieurs.

* *Le modèle est basé sur des unités de groupe de mots ( phrase )*
* Chaque groupe de mots vietnamiens correspond à un groupe de mots anglais , est la probabilité de déplacement de groupe de mots
* Les groupe de mots peuveut être déplacés, est la probabilité de déplacement
  + : première position du groupe de mots correspondant à
  + : première position du groupe de mots anglais correspondant à

La meilleure probabilité de déplacement:

* **Décoder**

Corpus après les 2 derniers modèles : langages et traduction va créer le table de probabilités pour chaque paramètre respectif. Le problème de trouver P (e) P (v|e) au maximum.

Deux algorithmes et un algorithme d'optimisation pour le modèle de recherche: algorithme glouton de recherche, l'algorithme de recherche basé sur la pile, et l'algorithme recherche le cycle optimal Hamilton. Deux algorithmes ont du temps de traitement plus rapide que l’algorithme, mais les résultats sont moins.

## 4.4. Utiliser Moses pour le probème d’ajouter les signes Vietnamiens

Pour ajouter les signes à un texte non signées vietnamiens, nous considérons que le vietnamien non signés comme une nouvelle langue, après l’entraîneur de la traduction automatique Moïse avec corpus bilingues (le vietnamien non signés – le vietnamien accentués) nous prenons le texte non signé en vietnamien à l'entrée, après le processus de traduction, on va obtenir le texte vietnamien accentué.

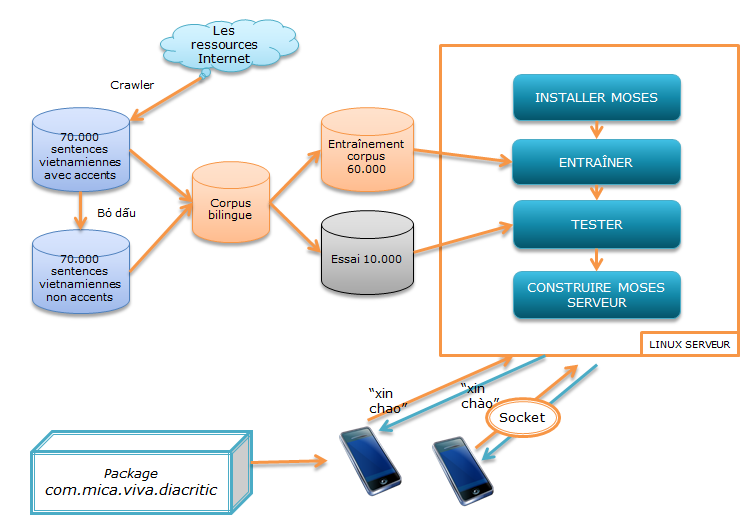


Figure 4.2 Modèle d’application de traduction automatique pour le problème d’ajouter les signes

Le déploiement de systèmes de traduction automatique Moses pour ajouter les signes vietnamiens est réalisé à travers les étapes suivantes:

### 4.4.1. Étape 1: Installer Moses Toolkit.

Moses fonctionne le mieux sur une plate-forme Unix. Pour exécuter le Moses, les logiciels doiveut être installés:

* Moses.
* GIZA++, pour aligner le goupe de mots dans les paires de phrases dans le corpus bilingue.
* IRSTLM, pour construire le modèle de langue.

### 4.4.2. Étape 2: Créer corpus.

Tout d'abord, pour construire corpus de pharses vietnamiennes accentuées, j’ai créé un Clawrer qui obtient les article en ligne tirée des médias en ligne se cassent contenu vietnamien de documents HTML recueillies. Corpus recueilli environ 70.000 phrases vietnamiennes (23 MB texte).

Après le corpus des phrases vietnamiennes, la créaction le corpus de vietnamien non signé est facile en remplaçant les lettres accentuées par les lettres non signé correspondant par la loi

Après cette étape, nous obtenons 2 corpus parallèles: vietnamese.ton, Vietnamese.noton, chaque corpus a été divisé en deux parties: une partie se compose de 60.000 (vietnamese-60k.ton, vietnamese-60k.noton) pour rendre les données formation, une autre partie 10.000 phrase restant pour servir le test vietnamese-10k-source.txt, vietnamese-10k-test.txt).

### 4.4.3. Étape 3: Entraîner le modèle de langue.

Modèle de langue cible pour s'assurer que la sortie correcte de la syntaxe de la langue. Outils IRSTLM nous aider à créer des modèles de langue pour le vietnamien accentué (la langue cible) facilement sur ​​le corpus collectif disponible avec quelques lignes de commande :

Après cette étape, nous obtenons le modèle de la langue pour le vietnamien accentué dans le fichier vietnamese.arpa.ton

### 4.4.4. Étape 4: Entraîner le système

Dans cette étape, nous allons utiliser le corpus a été élaboré pour former le système de traduction automatique. Pendant le processus d’entraînement, la fonction mot-alignement formation est réalisée avec GIZA + + outil, les phrases sont séparées et pondérés (scoring), créé le lien de vocabulaire, la table de modification de postion de mots.

Après cette étape, nous obtenons fichier Moses.ini contient les informations de configuration du système et les fichiers *phrase-table.gz, reodering-table.wbe-msd-bidirectional-fe.gz* contiennent les informations extrait après le processus de formation.

### 4.4.5. Étape 5: Contrôler:

Après l'étape de la formation, nous pouvons exécuter le programme à partir du fichier moses.ini reçu et les résultats testés.

Système de traduction automatique Moses était opérationnel et les résultats des tests traduit sont assez bons: d'ajouter les signes à la phrase “xin chao tat ca moi nguoi” 🡪 “xin chào tất cả mọi người”

## 4.5. Déploiement le module d'ajouter les signes au texte pour VIVA.

Système de traduction automatique Moïse a été mise en œuvre avec succès sur le serveur LINUX de l’institut MICA, le problème d’ajouter les signes a été résolu. Toutefois, étant donné l’ensemble de système dans l'application VIVA sur le téléphone n'est pas possible, parce que le système utilisé pour exécuter un grand nombre de ressources de mémoire (environ 8 GB de RAM), de sorte à pouvoir appliquer ce système à ajouter les signes au contenu du message dans les applications VIVA, j’ai construit un serveur de tracduction Moses qui permet au client de se connecter, envoyer des textes non signé en vietnamien par socket TCP / IP, serveur Moses effectuera la traduction (ajouter les signes ) et renvoyera du texte vietnamien accentué

### 4.5.1. Créer le serveur de traduction

Serveur de traduction de Moses est démarré par un simple segment Perl Script, ce script permet d'entrer deux paramètres lors de l'exécution le nom du serveur et le numéro de port.

### 4.5.2. Construire le module d’ajouter dans VIVA

Pour se connecter au serveur de traduction de Moses, dans l’application VIVA, nous construisons supplémentairement package com.mica.viva.diacritic qui contient classe VietnameseDiacritic pour se connecter au serveur Moses, envoyer la phrase qui a besoin d’ajouter les signes et recevoir la phrase a été ajouté.

## 4.6 Estimation du système

Pour évaluer le taux d'exactitude du système de la traduction Moses avait construit pour les signes supplémentaires pour le texte vietnamien, j'utilise 2 fichiers extraits du corpus: le fichier se compose de 10.000 phrases vietnamiens non signés non signés, dont 10.000 marqué le fichier vietnamien accentués respectivement. Le fichier non signé vietnamese-10k-test.txt traduits par Moses aux résultats du système de fichier vietnamese-10k-result.txt. Ce fichier de résultat est comparé avec le fichier de source vietnamese-10k-source.txt.

### 4.6.1 Estimation par la comparaison Word-By-Word.

Cet outil de comparaison est écrit dans Windows avec les fonctions suivantes:

* Comparer chaque ligne, respectivement, dans chaque ligne, chaque mot correspondant compter le nombre total de mots ajoutés les signes exactement, le nombre total de mots comparés, le nombre total de lignes.
* Caculer le taux de traduction correcte: le taux de traduction correcte = (nombre de mots corrects) / (nombre total de mots) \* 100 (%).
* Enregistrer le fichier de résultat et écrire les mots ajoutés faux avec leurs position, le numéro de ligne, position de mot sur cette ligne
* **Résultat**: te taux de précision: ~ 99,5%

**4.6.2. Estimation par le point BLEU.**

La valeur Score BLEU évalue la correspondance entre les deux traductions est effectuée sur chaque segment, le segment est ici comprise comme l'unité minimale de la traduction, chaque segment est généralement une phrase ou un période. Les statistiques coïncident carte des n-grammes basés sur un ensemble de n-grammes dans le secteur, d'une part elle est basée sur chaque segment, et puis re-calculer cette valeur dans tous les segments.

* **Résultat:** après l'exécution de commande pour appeler les outils à caculer le point bleu pour 2 fichiers, le fichier original et le fichier de résultat de traduction, l'outil affiche le bleu à 97.49.
* **Conclusion:** Depuis deux méthodes d'évaluation objectives, les résultats de l'évaluation montrent un taux d'exactitude de la méthode ajouter les signes du système de traduction automatique basé sur Moses est très élevé (plus de 99%), c'est le taux le plus élevé de précision a été annoncée par ce problème. Avec cette note, pour le système de traduction de corpus d'apprentissage n'est pas trop grand (~ 20MB 70000 question de texte), nous pouvons attendre qu’avec les grands corpus, de nombreux domaines et de sujets en plus de contenu (uniquement le contenu actuellement articles provenant du réseau, le manque de contenu tels que les conversations de chat, sms, email, commentaire ...), la qualité du système seront plus élevés.

# PROGRAMME VIVA DE VERSION DEMO

## 5.1 Joindre des modules

Après que les membres de l'équipe ont fait complètement les modules de VIVA (présenté dans la section 2.1), ils sont joins dans un programme complet:

* Programme VIVA est emballé dans un fichier VIVA.apk peut-être installé et exécuté sur des machines installées Android version (2.3.3) ou plus.
* Le module de reconnaissance, le module de compréhension sont mis sur le serveur Unix adressé 172.16.31.78 à l'Institut MICA. L’application VIVA sur les machines des clientes se connecte au service de reconnaissance et au service de compréhension sur ce serveur via le port 1569 et l’adresse MICA.edu.vn
* Le système de traduction automatique Moses fournit les services de livraison dans les documents texte vietnamiens dans bien marquées sur ce serveur sur le port 1571.

## 5.2 Demo du programme VIVA

Le groupe a fait l'installation et mise le programme VIVA sur deux appareils Android, compris le portable Google Nexus One (Android 2.3.6), Galaxy Tab 7 tablette (Android 2.3.3) et le portable Galaxy Note 2 (Android 4.1), les fonctions fonctionnent bien.

# RÉSUMÉ ET ÉVALUATION

## 6.1 Conclusion générale

Dans l'ensemble, le groupe a réussi à construire VIVA avec la version de demo, le programme a pu écouter les utilisateurs, comprendre ce que les utilisateurs veulent, et exécuter la fonction de réponse des utilisateurs par la voix. Personnellement, dans ce mémoire de fin d’études, j'ai présenté un système d'aperçu VIVA, la base théorique et basé sur l'architecture des blocs, ainsi que le détail des travails que j'ai fait: conception & construction de la structure de base de VIVA architecture, construction de module de fonction intégrée de messagerie, la recherche, la résolution le problème d’ajouter les signes pour le texte vietnamien non signé avec des résultats de haute précision, et appliquer les résultats de la fonction de lecture messages entrants de VIVA.

## 6.2 Orientation de développement

* Construire de nouvelles fonctions (section 1.3.3), particulièrement la fonction de conversation avec l'utilisateur, pour que le programme soit plus chic et amicale.
* Reclasser et vaincre les problèmes limités :
* Bloc de reconnaissance de la parole:
* Augmentation de la vitesse d'identification:
  + Élargir de base de données de formation: enregistrement plus de phrases, sur plusieurs autres sujets, l'augmentation du nombre de personnes, de matériels d'enregistrement, de contexte.
  + Augmentation de la vitesse d'identification:
    - Déplacer la partie de traitement et de sélection des caractéristiques sur les applications du portable, afin de réduire la manipulation de la charge du serveur tout en réduisant la taille des paquets transmis par socket.
    - Créer de multiples serveurs ou threads s'exécutent le bloc de reconnaissance parallèlement, le module Service Provider sera distribué au client se connecte au serveur ou suite logiquement afin d'accélérer la réponse du système.
* Le bloc de compréhension de commandes:
  + Augmentation de la flexibilité pour ce bloc par le développement un automate dynamique dont les états, les mots clés sont stockées dans une base de données ou dans le fichier (. txt,. xml) séparément avec l’algorithme dans le code .
  + Développer plus de fonction de compréhension des paramètres de contenu, de choix de l'utilisateur: par exemple, le système demande à l'utilisateur d'entrer le nom d'un destinataire du message, VIVA peut comprendre le contenu "à mon ami nommé Binh" c’est-à-dire que l’utilisateur veut envoyer un message à « Binh » ou «oui», «bien sûr», «je veux», «d’accord» est la sélection «oui» pour confirmer.
* Bloc de synthèse de la parole
  + Augmentation de la capacité de personnaliser la fréquence F0 pour la voix plus expressif
  + Augmentation de base de données pour que la synthèse soit plus précise et afin d’avoir plus de voix
  + Développer le module de normalisation de texte d’entrée avant la synthèse: comment lire les nombres, devises, date, nom, symboles abrégés ...
* Bloc pour ajouter les signes au texte vietnamien non signé :
  + Eslargir corpus de formation: ajouter les contenus avec le sujet de conversation, SMS, Email, chat, comment.

Liste des références

[1] Philipp Koehn, Franz Josef, Daniel Marcu. “Statistical Phrase-Based Translation”, Conference of the North American, Chapter of the Associat ion for Computational Linguist ics on Human Language technology - Volume 1, 2003

[2] Tufiş, D., Chiţu, A. “Automatic Insertion of Diacritics in Romanian Texts”. Proceedings of the 5th Int ernat ional Workshop on Comput at ional Lexicography COMP LEX, P ecs, Ungaria, 1999,185-194

[3] Rada Mihalcea, Vivi Nastase, “Letter Level Learning for Language Independent Diacrit ics Restoration ”, In: P roceedings of CoNLL 2002.

[4] Simard, M. “Automatic Insertion of Accents in French Texts”. In Ide & Vuotilainen (eds) P roceedings of the Third Conference on Empirical Methods in Natural Language Processing, Granada, Spain, 27-35, 1998

[5] Luu Tuan Anh, Kazuhide Yamamoto , “A P oint wise Approach for

Viet namese Diacritics Restoration”, in proceedings of IALP, 2012

[6] Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison- Burch, Richard Zens, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen and Christ ine Moran. “Moses: Open Source Toolkit for Statistcal Machine Translation”. Proceedings of the ACL 2007.