Reconnaissance vocale

Introduction

Dans le cadre du projet DLL on a la tâche d’étudier le développement d’un programme de reconnaissance vocale sur la plateforme FirefoxOS. Cette application doit pouvoir s’intégrer aux diverses applications existantes sur la plateforme. Elle doit par ailleurs s’exécuter en local et présenter un degré de fiabilité élevé.

Qu’est-ce que la reconnaissance vocale ?

La reconnaissance vocale, prise très généralement est l’association d’un signal sonore à un texte, la reconnaissance, d’une série de mots prononcés par un interlocuteur choisi était disponible dès 1950 [1]. Malheureusement à défaut de perspectives immédiates et faute de subventions la recherche est très vite arrivée à un point mort [2].  
C’est uniquement à partir de 1980 que la recherche a réellement progressé dans le domaine, suite aux utilisations des Modèles de Markov cachés (HMM) [3] puis l’association de ses derniers avec les réseaux de neurones [4] [5].

De quoi est composée une reconnaissance vocale ?

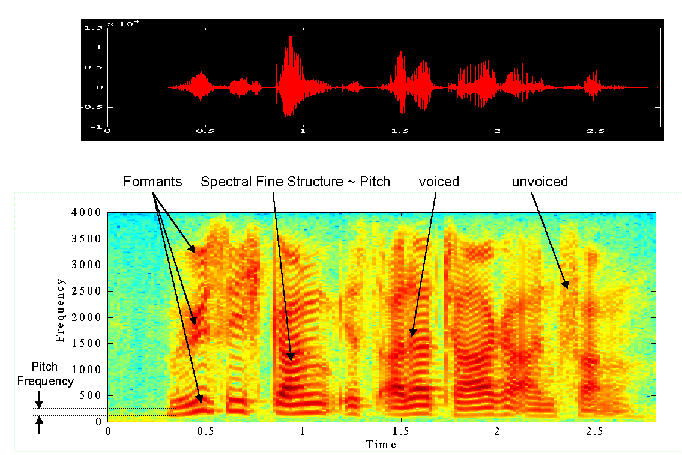
La reconnaissance vocale moderne nécessite plusieurs modules pour être entièrement opérationnelle. Nous nous baserons autours de PocketSphinx [6] dans ce document. Les raisons de ce choix sont le model opensource de l’outil et une certaine communauté formée autours, chose requise pour un logiciel de reconnaissance vocale.

Enregistrement du son

L’application doit pouvoir écouter la voix et avoir la capacité d’enregistrer cette dernière. Souvent un tel enregistrement doit subir de nombreux post-traitements pour assurer un environnement de reconnaissance idéal. Les environnements dans lesquels se passent les enregistrements ressemblent rarement aux chambres anéchoïques [7], il est nécessaire d’enlever le bruit de l’environnement et ramener la voix à une sonorité la plus neutre possible.

Bien qu’une personne ne prononce pas les mots toujours de la même manière selon son contexte (la personne peut être enrhumée, chuchoter, parler un peu plus vite ou plus lentement), il est possible de décomposer un langage en une série de phonèmes, le français en possède 36 dans sa version classique [8].

Décomposition du son

La décomposition du son en phonèmes ne peut se faire au format dans lequel il est initialement enregistré. Il est nécessaire de transformer le son, en une série de spectrogrammes [9] prises toutes les 1/25 – 1/50 de seconde [10]. Dans notre cas la partie décomposition est gérée par PocketShinx. Il est nécessaire que le son soit mono et dans un format 16KHZ [11] (valeur par défaut des modèles fournis). La transformation en spectrogrammes est généralement faite grâce à la transformation de Fourier rapide [12]. L’utilisation des spectrogrammes est plutôt fiable : pendant des années il y avait l’idée d’apprendre aux personnes malentendantes à les reconnaitre [13].   
Chaque frame est soumise à comparaison aux phonèmes reconnus du model de langage. Et le modèle fourni une série de «trouvés » potentiels à chaque touche. Dans le cas de WebApi Firefox (qui implémente PocketSphinx) un seul mot est retourné par touche [14]    
  
  
Bibliographie :

1. <http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf>
2. [Pierce, John R.](https://en.wikipedia.org/wiki/John_R._Pierce) (1969). "Whither speech recognition?"
3. <https://fr.wikipedia.org/wiki/Mod%C3%A8le_de_Markov_cach%C3%A9>
4. <https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels>
5. <http://static.googleusercontent.com/media/research.google.com/fr//pubs/archive/38131.pdf>
6. <http://cmusphinx.sourceforge.net/2015/12/pocketsphinx-wrappers-with-swig/>
7. <https://fr.wikipedia.org/wiki/Chambre_an%C3%A9cho%C3%AFque>
8. <https://fr.wikipedia.org/wiki/Phon%C3%A8me>
9. <https://en.wikipedia.org/wiki/Spectrogram>
10. <http://www.explainthatstuff.com/voicerecognition.html>
11. <http://cmusphinx.sourceforge.net/wiki/tutorialam>
12. <https://fr.wikipedia.org/wiki/Transformation_de_Fourier_rapide>
13. <https://www.researchgate.net/publication/221488179_Dynamical_spectrogram_an_aid_for_the_deaf>
14. <https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API>