

Projet Parole

Reconnaissance et Compréhension de la parole

Déroulement : 2 séances de 2h encadrées + travail personnel + validation machine

Travail à réaliser en binôme

Objectif : faire évoluer et intégrer les deux composantes développées en TP (TP5 et TP7) pour réaliser une application complète et l'évaluer.

Extensions envisagées :

- Effectuer un traitement en « temps réel » sans intervention humaine,
- Traiter d'autres opérateurs de calcul : +, -, *, /, %, ^, etc.
- Enchaîner plusieurs opérations. Exemple : $5+3*5$.
- Fournir le résultat de manière plus conviviale. Exemple : synthèse vocale de Google ou `espeak` (<http://espeak.sourceforge.net>),
- Traiter aussi des nombres en plus des chiffres modélisés dans le TP5 (10...99).

Le but est de réaliser une **calculatrice vocale**. L'utilisateur énonce une opération et le système lui fournit le résultat de l'opération par affichage et par synthèse vocale.

1) La **méthode de reconnaissance** utilisée est fondée sur les Modèles de Markov Cachés (Hidden Markov Models – HMM). Vous utiliserez pour cela la boîte à outil `HTK` (Hidden ToolKit distribué par <http://htk.eng.cam.ac.uk/>) et procéderez sur le même mode que pour le TP5.

Signal à traiter : signal direct avec utilisation de la commande `adinrec` ou via un fichier Wave

Sortie : suite de mots transcrite

Code : un script Unix lancera toute la chaîne de reconnaissance et produira la chaîne de caractères transcrite

2) La **méthode de compréhension** utilisée sera fondée sur une approche par règles (grammaire) et utilisera la librairie `NLTK` (<http://www.nltk.org/>). Vous procéderez de la même manière que pour le TP7.

Entrée : transcription de l'opération obtenue grâce au module de reconnaissance

Sortie : Représentation interne de l'opération à exécuter (opérateur, opérandes, niveau d'imbrication des opérations...)

Code : un script `Python` fera le travail d'analyse et générera l'interprétation de la commande reçue en entrée

Exemple :

Ainsi pour l'opération reconnue '**quinze plus vingt**' on obtiendra soit

```
[INTERP = [ OPERATION = add    OPERANDE1 = 15 OPERANDE2 = 20]]
```

ou

```
[INTREP = add(15, 20)]
```

ATTENTION : ce résultat intermédiaire devra être affiché à l'écran comme trace afin de permettre la vérification du bon fonctionnement du module.

EXTENSION : vous envisagerez la possibilité de traiter des opérations du type :

'cinq plus dix fois trois' 'vingt multiplié par trois fois deux'

3) A partir du résultat du module de compréhension, le **module calcul** exécutera l'opération et fournira le résultat selon 2 modalités : affichage écran ET synthèse vocale.

Entrée : résultat du module d'analyse

Sortie : Résultat du calcul

Code : le code sera écrit en python.

Application complète : un script Unix sera lancé et toute la chaîne exécutée.

4) Evaluation du projet – Code commenté et validation machine

Pas de rapport à rendre, mais quelques contraintes à respecter lors de la validation machine. Le système sera évalué de plusieurs manières différentes : (1) par composante et (2) globalement sur plusieurs types de données de test et (3) sur la qualité (lisibilité et commentaire) du code :

- sur vos enregistrements (**contexte multi-locuteurs**) pour valider votre système,
- sur des enregistrements inconnus (**contexte indépendant du locuteur**) pour connaître le potentiel d'adaptation de votre méthode,
- sur le potentiel de la partie compréhension (**couverture et flexibilité du langage**).

Ceci signifie que ce module ne doit pas être contraint par les limites du système de reconnaissance. Par exemple, si coté reconnaissance vous n'avez pas pris en compte le nombre '**quatre-vingt douze**', il ne doit pas pour autant être exclus de la partie analyse. Le module pourra être évalué en simulant le résultat du système de reconnaissance et en donnant directement le texte ('**quatre-vingt douze au carré**') de l'opération à exécuter.

5) Calendrier

Une **validation sur machine** est prévue lors de la dernière séance. Cette validation d'une quinzaine de minutes permettra de vérifier que vous avez bien atteint les objectifs fixés par ce projet. **Votre code sera également déposé sur Moodle.**