

Le système de dialogue

Une introduction rapide

Sophie Rosset

Groupe Traitement du Langage Parlé
Département Communication Homme-Machine
LIMSIS - CNRS

1 Idial

2 ex_dialogue.lua

3 Les différents appels

Idial : c'est quoi ?

dialogue et lua

Pas à pas

Idial ?

- Infrastructure de calculs distribués
- Système communicant
- Plate-forme pour système de dialogue (et d'autres !)
- Wmatch en natif

→ fonctionne avec des scripts lua

Idéal : c'est quoi ?

dialogue et lua

Pas à pas

ex_dialogue.lua : les différentes sections

- Appels des différents fichiers via des *dofile*
- Interface utilisateur : lancement de la première intervention du système
- Chargement du système
- envoi des entrées utilisateur au système

Idéal : c'est quoi?

dialogue et lua

Pas à pas

Exemple simple de fonction

C'est très proche du pascal...

```
function liste_has_value(list,value)
    local i,n
    for i,n in ipairs (list) do
        if (n==value) then
            return true
        end
    end
    return false
end
```

Idéal : c'est quoi ?

dialogue et lua

Pas à pas

Système : master.lua

- Initialisation de l'analyse, de la tâche, définition d'une typologie des différents éléments utiles
- Traitement des énoncés utilisateurs et des réactions système

Non contextual analysis : nca.lua

- fonction `nca_init()` : charge l'analyseur indiqué dans le path
- fonction `nca_run(sentence)` : passe l'analyseur sur l'énoncé en entrée

Parcours d'arbres : tree.lua

Différents fonctions pour parcourir les arbres de l'analyse, extraire des valeurs et des attributs

Idéal : c'est quoi ?

dialogue et lua

Pas à pas

Dialogue : dialogue.lua

- fonction guide : Si réponse "oui" et que système a pas encore répondu, alors c'est une demande de guide.
- fonction dial_db : gestion historique - variables locales pour l'historique et globale pour la vérification des valeurs présentes dans l'analyse.
- Exemple d'application au pays...

Idéal : c'est quoi ?

dialogue et lua

Pas à pas

Historique PAYS

- déclaration des variables locales
 - local info=extract_tag_multi(utterance,liste_tag_rechdb)
 - local liste_T = "_Tcapitale"
- création de l'historique
 - hist_rechdb :store("info_hc", info)
 - local info_prec=hist_rechdb :get(-1).info_ec
 - extract_tag_multi(utterance,liste_tag_rechdb) : extraction des éléments de la liste qui sont dans l'énoncé courant → info
 - hist_rechdb :store("info_hc", info) : mettre info dans info hors contexte
 - info_prec=hist_rechdb :get(-1).info_ec : mettre dans info_prec ce qui était dans info_ec du tour précédent

Application

- Si on a une info précédente (info_prec) alors
- Récupérer liste_T
- Si on a pays et pas de liste_T et un liste_T ancien alors
- Récupération des éléments de liste_T d'avant dans le info courant
- Sinon si pas d'info_pays mais un liste_T et un ancien pays alors
- Récupération de l'ancien pays dans pays

J'ai dans l'historique une demande sur Tcapitale, dans l'énoncé courant j'ai un pays, je récupère Tcapitale et je gère Tcapitale + pays.

J'ai dans l'historique un pays, dans l'énoncé courant Tcapitale et pas de pays, je récupère l'ancien pays et gère Tcapitale + pays

Idial : c'est quoi ?

dialogue et lua

Pas à pas

Idéal : c'est quoi ?

dialogue et lua

Pas à pas

Recherche db

Si Tcapitale et pays alors

```
local rechcapitale=caps:select{pays=info._pays[1]}
  if (rechcapitale:line_count())<1) then
```

alors on répond pays_inconnu

```
  fin
```

```
  return
```

```
{acte=" reponse" ;content=" capitale" ;pays=info._pays[1]...
```

```
  fin
```

Si Tcapitale et pas pays alors

On répond paspays