

Les quatre patterns à connaître pour faire reconnaître un "mot clé unique"

Soumis par Philippe YONNET

11-05-2008

Dernière mise à jour : 11-05-2008

Les quatre patterns à connaître pour faire reconnaître un "mot clé unique" dans la plupart des cas.

<pattern>MOT</pattern> (matche avec "mot")

<pattern>MOT *</pattern> (matche avec "mot doux")

<pattern>* MOT</pattern> (matche avec "tu veux avoir le dernier mot ?")

<pattern>* MOT *</pattern> (matche avec "tu as le mot pour rire !")

Attention, toutefois, il est possible que "mot" ne soit pas reconnu s'il vient à être zappé parce qu'une autre partie du stimulus matche avec une expression prioritaire

Si vous voulez que cela n'arrive pas (en tout cas le moins possible) essayez ce qui suit. Les quatre patterns à connaître pour faire reconnaître un "mot clé unique" dans 90% des cas.

<pattern>MOT</pattern> (matche avec "mot")

<pattern>MOT *</pattern> (matche avec "mot doux")

<pattern>_ MOT</pattern> (matche avec "tu veux avoir le dernier mot ?")

<pattern>_ MOT *</pattern> (matche avec "tu as le mot pour rire !")

Notez l'ajout des underscores au début...

Attention toutefois, à présent ce sont d'autres parties du stimulus qui seront zappées, la réponse sur "mot" étant traitée en priorité. A utiliser à bon escient, il convient absolument dans ce cas de rédiger un template avec un <star/> à l'intérieur au cas où.

Pourquoi pas de "_" à la fin et uniquement au début ?

Les stimulus étant analysés de gauche à droite, cela permet de faire reconnaître "mot" en priorité en "shuntant" tout ce qui était situé avant le terme "mot" dans le stimulus et qui matchait avec d'autres patterns, mais en laissant la possibilité d'avoir de multiples réponses pertinentes en fonction de ce qui suit "mot". Avec un _ à gauche et à droite, une seule réponse est possible quel que soit le stimulus contenant "mot".

Les 3 mesures à prendre pour éviter la "collision" entre deux catégories

1. Tester, tester, et retester le chatterbot, soi même mais aussi en faisant travailler d'autres personnes. Les logs de Pandorabots peuvent également être utiles
2. Ne jamais créer de patterns à l'aide d'expressions ambiguës, ou pouvant prendre plusieurs sens, ou "recombinables" avec d'autres termes (du moins si possible). S'il n'y a pas moyen d'y parvenir, créer des catégories ou des templates permettant une désambiguation. Mais ce n'est pas toujours aisé, je le concède.
3. Bien hiérarchiser les patterns.