

Quand utiliser "star"=* et underscore="_" ?

Soumis par Philippe YONNET

11-05-2008

Dernière mise à jour : 11-05-2008

Rappelons l'exemple déjà utilisé dans l'article précédent : d'un côté, des catégories chargées de répondre aux insultes.

```
<category>
<pattern>ESPECE DE *</pattern>
<template>Malotru !</template>
</category>
```

```
<category>
<pattern>ESPECE D *</pattern>
<template>Goujat !</template>
</category>
```

```
<category>
<pattern>_ ESPECE *</pattern>
<template>Mal élevé !</template>
</category>
```

```
<category>
<pattern>_ ESPECES D *</pattern>
<template>Toi-même !</template>
</category> De l'autre, on cherche à donner des infos sur les ornithorynques :<category>
<pattern>ORNITHORYNQUES</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>ORNITHORYNQUE</pattern>
<template>L'ornithorynque (Ornithorhynchus anatinus) est une petite espèce de mammifère semi-aquatique endémique
vivant dans l'est de l'Australie</template>
</category>
```

Et imaginons que quelqu'un pose la question suivante :

"Est-ce qu'il y a plusieurs espèces d'ornithorynque ?". Je pense que la réponse "toi même !" risque de surprendre l'utilisateur...

(la phrase ci dessus correspond au pattern _ ESPECES D *)

En fait il y'a une collision entre deux patterns, le premier "_ ESPECES D *" prenant le dessus.

Une phrase commençant par "espèce de" est probablement une insulte. On peut garder les deux premières catégories telles quelles. Une phrase avec le mot "espèce" au milieu est ambiguë. On peut soit supprimer ce pattern pour laisser le chatterbot répondre de façon générique quelque chose du genre "je ne sais pas si les ornithorynques sont une espèce en danger" ou simplement "je ne sais pas".

Ensuite on peut soit forcer la reconnaissance du terme ornithorynque dans tous les cas, en utilisant le joker underscore (souligné) en début de chaîne :

```
<category>
<pattern>_ ORNITHORYNQUE</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>ORNITHORYNQUE *</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
```

```
<pattern>_ ORNITHORYNQUE *</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>_ ORNITHORYNQUES</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>ORNITHORYNQUES *</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>_ ORNITHORYNQUES *</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>ORNITHORYNQUES</pattern>
<template><srai>ORNITHORYNQUE</srai></template>
</category>
```

```
<category>
<pattern>ORNITHORYNQUE</pattern>
<template>L'ornithorynque (Ornithorhynchus anatinus) est une petite espèce de mammifère semi-aquatique endémique
vivant dans l'est de l'Australie</template>
</category>
```

Le fait de placer un underscore en début de chaîne change totalement l'ordre de vérification des patterns.

Il convient de se pencher quelques minutes sur une des propriétés fondamentales du langage AIML : l'ordre des priorités dans les analyses de correspondance avec les formes (patterns).

Dans le cas où l'on ne compose de patterns qu'à l'aide du joker "*", l'ordre de priorité est assez intuitif.

Les correspondances sont recherchées en lisant les stimulus de gauche à droite

La première des choses à savoir c'est que les stimulus sont analysés de gauche à droite.

Si on écrit ces deux patterns :

```
<pattern>* ORNITHORYNQUES *</pattern>
<pattern>* ESPECES D *</pattern>
```

Le stimulus

"Est-ce qu'il y'a plusieurs espèces d'ornithorynques en australie ?"

matche avec les deux patterns décrits ci-dessus. Comme il doit choisir entre les deux, l'ordre de priorité est fixé par l'ordre dans lequel la première correspondance avec un pattern existant est découverte : c'est donc le pattern "*" especes d*" qui sera choisi, et "*" ornithorynques *" sera superbement ignoré.

Avec des stimulus commençant par *, dès qu'un pattern correspond au début de la phrase, il est choisi !

"*" est un joker qui fonctionne par défaut : un pattern qui contient un "*" ne sera pris en compte que si aucun pattern contenant les termes exacts du stimulus n'a été trouvé avant.

```
exemple : <category>
<pattern>QUI EST *</pattern>
<template>Je ne sais pas qui est <star/></template>
</category>
```

```
<category>
<pattern>QUI EST VICTOR HUGO</pattern>
<template>Victor Hugo est un grand écrivain français du 19 siècle</template>
</category>
```

```
<category>
<pattern>QUI EST VICTOR *</pattern>
<template>Euh. Je connais Victor Hugo, mais pas ce victor là</template>
</category>
```

```
<category>
<pattern>QUI EST * HUGO</pattern>
<template>Je connais Victor Hugo, mais pas <star/></template>
</category>
```

Au stimulus "Qui est Alfred de Musset ?" la réponse est : Je ne sais pas qui est Alfred de Musset
 Au stimulus "Qui est Victor Schoelcher ?" la réponse est : Euh. Je connais Victor Hugo, mais pas ce victor là.
 Au stimulus "Qui est Victor Hugo ?" la réponse est : Victor Hugo est un grand écrivain français du 19 siècle.
 Au stimulus "Qui est Léopoldine Hugo ?" la réponse est : Je connais Victor Hugo, mais pas Léopoldine.

On voit bien la hiérarchie de gestion des réponses, qui permet de créer des règles par défaut en cascade : si le nom est connu, on donne la réponse exacte. Si une partie du nom est connue, on donne une réponse pertinente. Si le nom est inconnu, la règle par défaut est de répondre "je ne connais pas".

"*" permet de construire des règles par défaut en cascade. Les règles contenant "*" sont utilisées si une correspondance exacte n'est pas trouvée.

Dans la pratique, on construit les fichiers AIML dans l'autre sens : d'abord des patterns "catch all", des règles "attrape tout". Ensuite on crée des bases de connaissances précises.

Que se passe-t'il si je remplace les "*" par des "_" ?

```
Au stimulus "Qui est Alfred de Musset ?" la réponse est : Je ne sais pas qui est Alfred de Musset
Au stimulus "Qui est Victor Schoelcher ?" la réponse est : Je ne sais pas qui est Victor Schoelcher.
Au stimulus "Qui est Victor Hugo ?" la réponse est : Je ne sais pas qui est Victor Hugo.
Au stimulus "Qui est Léopoldine Hugo ?" la réponse est : Je ne sais qui est Léopoldine Hugo.
```

En fait, tous ces stimulus matchent à présent avec le pattern le plus court correspondant "QUI EST _". Les autres réponses ne peuvent plus apparaître.

La réponse "par défaut" est devenue une réponse "prioritaire".

IMPORTANT : n'utilisez de "_" que si vous voulez vraiment "prendre le dessus" sur d'autres patterns. Dans tous les autres cas, c'est inutile. Si quelque chose ne se passe pas comme vous le voulez, analysez l'AIML, et n'utilisez "_" qu'en dernier recours.

L'utilisation des "_" est légitime dans trois cas seulement :

- les règles de réduction
- le "court circuitage" de patterns existants si vous ne voulez pas changer les fichiers de base du chatterbot et rajouter quelques règles à vous
- les mots-clés qui doivent déclencher une réponse très précise

Nota bene : vous remarquerez peut-être que j'utilise beaucoup moins "_" dans mes jeux de fichiers AIML que ce que l'on trouve dans A.L.I.C.E. par exemple. En fait, c'est pour résoudre des problèmes qui viennent du français, qui est beaucoup moins elliptique et beaucoup plus bavard que l'anglais, et surtout, beaucoup moins "régulier" au niveau de l'ordre des mots (adjectif avant ou après, l'adverbe se place un peu n'importe où, les compléments directs ou indirects aussi et surtout il y'a bien cinquante façons de poser une question...). Inconvénients : il faut rédiger les fichiers de manière super logiques, sinon gare, et le graphe est un peu lourd à parser par les programmes de traitement. Mais dans la pratique cela marche très bien.