

百度文库：AIML规范 - 综合论坛 - AI 人工智能论坛 - Powered by Discuz!

星期日, 6月 9 2013, 1:59 下午

百度文库：**AIML**规范

Aiml就是一个xml兼容的文本文档，现简单介绍一下xml规范：

xml每一个型如<aiml topic=" 电影">叫做一个元素，每个元素都是闭合的，也就是说要只要有<aiml>开头就必定有</aiml>相对应与之闭合；topic=" 电影" 称为改元素的属性，一个元素可以有多个属性没有限制。元素和元素之间可以嵌套，但必须闭合。

一个最简单的aiml文件如下：

```
<?xml version="1.0" encoding="GB2312"?>
```

```
<aiml>
```

```
<category>
```

```
<pattern>你好</pattern>
```

```
<template>好</template>
```

```
</category>
```

```
</aiml>
```

1.

<?xml version="1.0" encoding="GB2312"?>这个是所有xml文档都必须声明的，如果是中文这里要声明编码为：GB2312，如果是英文则一般声明为：UTF-8

2.

category表示一个**目录**，表示一个一问一答匹配和一问多种应答匹配，但不允许多中提问匹配。

3.

pattern表示**匹配**模式，表示用户的输入匹配，以上例子，用户一旦输入你好，那机器人就找到这个匹配，然后取出应答“好”；

4.

template表示**应答**，这里应答一个“好”字。

有了这几个简单的元素理论上就可以写出任意匹配模式，达到一定智能，但实际应用当中只有这些元素是不够的，接下去将作进一步介绍。

注意：**aiml**所有元素属性都是大小写敏感的！！！！

aiml的pattern元素里**面的英文必须大写**！！！！

4、详细说明

1.

think元素

记忆功能

型如：<think><set name="topic">Me</set></think>

放置在template元素里面，表示一旦用户的输入匹配到该category时，再回复应答的同时，给一个变量赋值，这里也就是把Me记再脑子里，以后就可以用<get name=" topic" />来取出先前记住的内容。

2.

<star/>表示*****，比如有一个匹配模式是<pattern>* 你 好 *</pattern>；注意，由于机器人解析器是外国人写的，而外国人每个单词之间一个空格，因为，我们写匹配的时候也要再每个汉字之间加上空格。这里**pattern**元素里的匹配模式是用*****号表示任意匹配的，但在其他元素里面不能用*****号，而用**<star/>**这个元素来表示。

3.

智能

<srai>元素，刺激回复**人工智能**，表示<srai>里面的话会被当作是用户输入，从新查找匹配模式，直到找到非<srai>定义的回复。例如：

<srai>我是 <star/></srai>，那么机器人会把“我是*”当作是用户输入来从新查找匹配模式。

4.

<condition>元素，放在template元素里面，可以有多个condition元素，但不能嵌套（目前还不支持），有3种形式：

```
<condition name="name" value="value">你好 </condition>
```

```
<condition name="name" contains="value"> 你好</condition>
```

```
<condition name="name" exists="value">你好 </condition>
```

Name是预先定义的变量，第一种表示name变量的值如果和value相等，回复内容就包括”你好”；第二种表示name变量的值如果里面包含value这个字符串，回复内容就包括“你好”；第三种表示name变量的值如果存在value的值，回复内容就包括“你好”（这个什么意思还不明确）

举个使用例子：

```
<category>
```

```
<pattern>你 好</pattern>
```

```
<template>
```

```
你好阿！
```

```
<condition name="用户名字" value="张三">
```

```
你又来了阿：)
```

```
</condition>
```

```
</template>
```

```
</category>
```

a.

情况一：如果事先用think设置了变量“用户名字”，而且等于“张三”，那么用户接受到的回复内容是：你好阿！你又来了阿；)

b.情况二：如果事先没有定义或者变量“用户名字”的值不等于“张三”，那么用户接受到的回复内容是：你好阿！

写法有以下几种：

```
<category>
```

```
<pattern>我头发的颜色是蓝色*</pattern>
```

```
<template>哇塞，你很
```

```
<condition name="用户性别" value="女"> 漂亮阿！</condition>
```

```
<condition name="用户性别" value="男">英俊阿！</condition>
```

```
</template>
```

```
</category>
```

```
<category>
```

```
<pattern>我头发的颜色是蓝色*</pattern>
```

```
<template>哇塞，你很
```

```
<condition>
```

```
<li name="用户性别" value="女">漂亮阿！</li>
```

```
<li name="用户性别" value="男">英俊阿！</li>
```

```
</condition>
```

```
</template>
```

```
</category>
```

```
<category>
```

```
<pattern>我头发的颜色是蓝色*</pattern>
```

```
<template>哇塞，你很
```

```
<condition name="用户性别">
```

```
<li value="女"> 漂亮阿！</li>
```

```
<li value="男"> 英俊阿！</li>
```

```
</condition>
```

```
</template>
```

```
</category>
```

```

<category>
<pattern>我头发的颜色是蓝色 *</pattern>
<template>哇塞，你很
<condition>
<li name="用户性别" value="女">漂亮阿！</li>
<li>好看！</li>
</condition>
</template>
</category>

```

这里最后一个li元素表示默认的情况；即找不到condition相匹配时，就用默认的。

5. **<formal>**元素，用来格式化输出，例如：`<formal>jon baer</formal>`
那么回复将被格式化成首字母大写输出：Jon Baer，对中文无效。

6. **<gender>**元素，替换性别以及代名词，例如：

`<gender>She told him to take a hike.</gender>`

将被替换成：He told her to take a hike，跟性别有关的单词都将被替换，中文怎么处理不是很清楚。

7. **<get name=" " 名字/>**，即得到name的值。

8. **<gossip>**元素用来把改元素里面的内容保存到gossip.log文件里。

9. **<if>**元素，判断元素，有以下形式：

`<if name="topic" value="cars"></if>`

`<if name="topic" contains="cars"></if>`

`<if name="topic" exists="true"></if>`

例子：

`<template>`

`<if name="用户名称" exists="true">`

你的名字叫 `<get name=" 用户名称" />`。

`<else/>`

你叫什么名字？

`</if>`

`</template>`

10. **<input>**表示用户输入，例如：

`<input index="2"/>`将输出用户倒数第2次的输入，看如下对话：

用户：好阿

机器人：你也好，你叫什么名字？

用户：我叫黑山

机器人：呵呵，你好黑山。

如果有aiml文件里有片断：

`<category>`

`<pattern>嘿 嘿</pattern>`

`<template>`

`<gossip>你刚才说：“<input index="2"/>”？</gossip>`

`</template>`

`</category>`

如果这个时候用户输入“嘿嘿”，那么机器人将回复：你刚才说：“我叫黑山”？

Index属性的数字表示倒数第几句话，1：倒数第一句，2：倒数第2句，依此类推。

11. **<learn** 应该为learn filename="xxx.aiml">元素表示让机器人学习某个aiml文件。

12. ****元素可以在**random**元素或**condition**等元素当中使用；表是一个列表；

13. `<lowcase>`/`</lowcase>`表示把中间的内容变成小写，对应的是`<upcase>`把内容变成大写中文无效。
14. `<pattern>`表示匹配模式，里面的内容必须大写，可以有星号*或下划线_，但必须空格隔开，星号表示匹配所有，任意情况；下划线的意义跟星号一样，除了不能匹配字典里面Z后面的字母。

15. `<person>`和`<person2>` 第一个是把第一人称转换成第3人称；第2个是把第一人称转成第2人称。`<person/>`等于`<person><star/></person>`

16. `<random>`随机元素，一般和``一起使用，表示从列表里随机取一个。

17. `<sentence>`元素用来格式化句子，比如：

```
<sentence>this is some kind of sentence test.</sentence>
```

可以格式化成：This is some kind of sentence test.即把句子首字母大写。

18. `<star/>`表示星号，已经不推荐使用，只针对0.9版本，目前一般要加index属性来使用，例如：`<star index="1"/>`，看如下对话：

用户：我的名字叫黑山！

```
<pattern>* 名字 叫 *</pattern>
```

```
<template>呵呵，我知道了，你的名字叫<star index="2"/>。
```

```
</template>
```

19. `<system>``<system>`元素表示调用系统函数，例如：

```
<system>date</system>
```

表示取系统当前日期

警告：使用这个元素要千万小心，因为有可能使用户运行你的系统命令。

20. `<template>`定义回复的模板。

21. `<that>`元素表示先前机器人说的话，例如：

```
<category>
```

```
<pattern>好</pattern>
```

```
<that>一起聊聊电影好吗*</that>
```

```
<template>那你喜欢什么电影那？</template>
```

```
</category>
```

即：如果机器人先前问用户“一起聊聊电影好吗？”，而且现在用户回答了“好”，那么匹配正确，回复内容为：“那你喜欢什么电影那？”

如果要取前面的前面机器人的话，可以用：`<that index=" nx,ny">`，例如：`<that index=" 2,1">`

表示取机器人倒数第2句的话，`<that index=" 2,1">`也等于`<justbeforethat/>`

22. `<thatstar index=" n">`元素先前第n个机器人说过的星号部分的话，`<thatstar/>`等于`<thatstar index="1"/>`。看个例子：

```
<category>
```

```
<pattern>你好</pattern>
```

```
<template>
```

```
计算机的型号是什么
```

```
</template>
```

```
</category>
```

```
<category>
```

```
<pattern>*</pattern>
```

```
<that>*
```

```
的型号是什么</that>
```

```

<template><star/>
这个型号是<thatstar/>里面
<random>
<li>很好的商品</li>
<li>很流行的商品</li>
<li>很华丽的商品</li>
<random>。
</template>
</category>
对话场景：
用户：你好
机器人：计算机的型号是什么
用户：p4
机器人：p4这个型号是计算机里面很好的商品

```

23.<think>记录用户输入且里面的内容不回复给用户。例如：

```

<category>
<pattern>I AM FEMALE</pattern>
<template>Thanks for telling me your gender.
<think>
<set name=gender>female</set>
</think>
</template>
</category>
这里把female这个值保存到gender这个变量里，且回复内容不包括female。

```

24. <Topic name=" film" >元素用来设置主题，例如：

```

<aiml>
<topic name="功夫">

<category>

<pattern> phrase </pattern>

<that> phrase </that>

<template> phrase </template>

</category>
</topic>
</aiml>

```

只有当前的topic系统变量是“功夫”时才会匹配到这里。

25. <topicstar index=" n" >元素用来得到先前倒数第n次谈论的主题。

26. <upcase>xiaoxie</upcase>元素用来把xiaoxie转换成大写形式：XIAOXIE

本文来自CSDN博客，转载请标明出处：<http://blog.csdn.net/woowindice/archive/2005/02/25/302298.aspx>

