

AIML - <SRAI> TAG

http://www.tutorialspoint.com/aiml/aiml_srai_tag.htm

Copyright © tutorialspoint.com

<srai> Tag is a multipurpose tag. This tag enables AIML to define different targets for same template.

Syntax

```
<srai> pattern </srai>
```

Following are the commonly used terms associated with srai:

- Symbol Reduction
- Divide and Conquer
- Synonyms resolution
- Keywords detection

Symbolic Reduction

Symbolic reduction technique is used to simplify patterns. It helps to reduce complex grammatical patterns with simple patterns.

For example, consider following conversation.

```
Human: Who was Albert Einstein?  
Robot: Albert Einstein was a german physicist.  
Human: Who was Issac Newton?  
Robot: Issac Newton was a english physicist and mathematician.
```

Now What if questions are raised as

```
Human: DO YOU KNOW WHO Albert Einstein IS?  
Human: DO YOU KNOW WHO Issac Newton IS?
```

Here <srai> tag works. It can take the pattern of user as a template.

Step 1: Create categories

```
<category>  
  <pattern>WHO IS ALBERT EINSTEIN?</pattern>  
  <template>Albert Einstein was a german physicist.</template>  
</category>  
<category>  
  <pattern> WHO IS ISSAC NEWTON? </pattern>  
  <template>Issac Newton was a english physicist and mathematician.</template>  
</category>
```

Step 2: Create generic category using <srai> tag

```
<category>  
  <pattern>DO YOU KNOW WHO * IS?</pattern>  
  <template>  
    <srai>WHO IS <star/></srai>  
  </template>  
</category>
```

Example

Create `srai.aiml` inside **C > ab > bots > test > aiml** and `srai.aiml.csv` inside **C > ab > bots > test > aimlif** directories.

srai.aiml

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0.1" encoding="UTF-8"?>
  <category>
    <pattern> WHO IS ALBERT EINSTEIN </pattern>
    <template>Albert Einstein was a german physicist.</template>
  </category>
  <category>
    <pattern> WHO IS ISSAC NEWTON </pattern>
    <template>Issac Newton was a english physicist and mathematician.</template>
  </category>
  <category>
    <pattern>DO YOU KNOW WHO * IS</pattern>
    <template>
      <srai>WHO IS <star/></srai>
    </template>
  </category>
</aiml>
```

star.aiml.csv

```
0,WHO IS ALBERT EINSTEIN,* ,*,Albert Einstein was a german physicist.,srai.aiml
0,WHO IS ISSAC NEWTON,* ,*,Issac Newton was a english physicist and
mathematician.,srai.aiml
0,DO YOU KNOW WHO * IS,* ,*,<srai>WHO IS <star/></srai>,srai.aiml
```

Execute Program

Open command prompt. Go to **C > ab >** and type the following command:

```
java -cp lib/Ab.jar Main bot=test action=chat trace=false
```

Verify Result

```
Human: Do you know who Albert Einstein is
Robot: Albert Einstein was a german physicist.
```

Divide and Conquer

Divide and Conquer is used to reuse sub sentences in making a complete reply. It helps to reduce defining multiple categories.

For example, consider following conversation.

```
Human: Bye
Robot: GoodBye!
Human: Bye Alice!
Robot: GoodBye!
```

Now here robot is expected to reply **GoodBye!**, Whenever a user says **Bye** in the beginning of the sentence.

Let's put `<srai>` tag to work here.

Step 1: Create category

```
<category>
  <pattern>BYE</pattern>
  <template>Good Bye!</template>
</category>
```

Step 2: Create generic category using <srail> tag

```
<category>
  <pattern>BYE *</pattern>
  <template>
    <srail>BYE</srail>
  </template>
</category>
```

Example

Update srail.aiml inside **C > ab > bots > test > aiml** and srail.aiml.csv inside **C > ab > bots > test > aimlif** directories.

srail.aiml

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0.1" encoding="UTF-8"?>
  <category>
    <pattern> WHO IS ALBERT EINSTEIN </pattern>
    <template>Albert Einstein was a german physicist.</template>
  </category>
  <category>
    <pattern> WHO IS ISSAC NEWTON </pattern>
    <template>Issac Newton was a english physicist and mathematician.</template>
  </category>
  <category>
    <pattern>DO YOU KNOW WHO * IS</pattern>
    <template>
      <srail>WHO IS <star/></srail>
    </template>
  </category>
  <category>
    <pattern>BYE</pattern>
    <template>Good Bye!</template>
  </category>
  <category>
    <pattern>BYE *</pattern>
    <template>
      <srail>BYE</srail>
    </template>
  </category>
</aiml>
```

star.aiml.csv

```
0,WHO IS ALBERT EINSTEIN,*,*,Albert Einstein was a german physicist.,srail.aiml
0,WHO IS ISSAC NEWTON,*,*,Issac Newton was a english physicist and
mathematician.,srail.aiml
0,DO YOU KNOW WHO * IS,*,*,<srail>WHO IS <star/></srail>,srail.aiml
0,BYE,*,*,Good Bye!,srail.aiml
0,BYE *,*,*,<srail>BYE</srail>,srail.aiml
```

Execute Program

Open command prompt. Go to **C > ab >** and type the following command:

```
java -cp lib/Ab.jar Main bot=test action=chat trace=false
```

Verify Result

```
Human: Bye
Robot: GoodBye!
Human: Bye Alice!
```

```
Robot: GoodBye!
```

Synonyms Resolution

Synonyms are words with similar meanings. A bot should reply in same manner for similar words. For example, consider following conversation.

```
Human: Factory
Robot: Development Center!
Human: Industry
Robot: Development Center!
```

Now here robot is expected to reply **Development Center!** whenever a user says **Factory** or **Industry**.

Let's put `<srail>` tag to work here.

Step 1: Create category

```
<category>
  <pattern>FACTORY</pattern>
  <template>Development Center!</template>
</category>
```

Step 2: Create generic category using `<srail>` tag

```
<category>
  <pattern>INDUSTRY</pattern>
  <template>
    <srail>FACTORY</srail>
  </template>
</category>
```

Example

Update `srail.aiml` inside **C > ab > bots > test > aiml** and `srail.aiml.csv` inside **C > ab > bots > test > aimlif** directories.

srail.aiml

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0.1" encoding="UTF-8"?>
  <category>
    <pattern> WHO IS ALBERT EINSTEIN </pattern>
    <template>Albert Einstein was a german physicist.</template>
  </category>
  <category>
    <pattern> WHO IS ISSAC NEWTON </pattern>
    <template>Issac Newton was a english physicist and mathematician.</template>
  </category>
  <category>
    <pattern>DO YOU KNOW WHO * IS</pattern>
    <template>
      <srail>WHO IS <star/></srail>
    </template>
  </category>
  <category>
    <pattern>BYE</pattern>
    <template>Good Bye!</template>
  </category>
  <category>
    <pattern>BYE *</pattern>
    <template>
      <srail>BYE</srail>
    </template>
  </category>
```

```

    </template>
</category>
<category>
  <pattern>FACTORY</pattern>
  <template>Development Center!</template>
</category>
<category>
  <pattern>INDUSTRY</pattern>
  <template>
    <srai>FACTORY</srai>
  </template>
</category>
</aiml>

```

star.aiml.csv

```

0,WHO IS ALBERT EINSTEIN,* ,*,Albert Einstein was a german physicist.,srai.aiml
0,WHO IS ISSAC NEWTON,* ,*,Issac Newton was a english physicist and
mathematician.,srai.aiml
0,DO YOU KNOW WHO * IS,* ,*,<srai>WHO IS <star/></srai>,srai.aiml
0,BYE,* ,*,Good Bye!,srai.aiml
0,BYE * ,*,*,<srai>BYE</srai>,srai.aiml
0,FACTORY,* ,*,Development Center!,srai.aiml
0,INDUSTRY,* ,*,<srai>FACTORY</srai>,srai.aiml

```

Execute Program

Open command prompt. Go to **C > ab >** and type the following command:

```
java -cp lib/Ab.jar Main bot=test action=chat trace=false
```

Verify Result

```

Human: Factory
Robot: Development Center!
Human: Industry
Robot: Development Center!

```

Keywords Detection

Using srai, we can accomplish a simple response if user types a specific keyword say School no matter where school is present in the sentence.

For example, consider following conversation.

```

Human: I love going to school daily.
Robot: School is an important institution in a child's life.
Human: I like my school.
Robot: School is an important institution in a child's life.

```

Now here robot is expected to reply a standard message **School is an important institution in a child's life**. whenever a user have **school** in the sentence.

Let's put <srai> tag to work here. We'll use wild cards here.

Step 1: Create category

```

<category>
  <pattern>SCHOOL</pattern>
  <template>School is an important institution in a child's life.</template>
</category>

```

Step 2: Create generic categories using <srai> tag

```

<category>
  <pattern>_ SCHOOL</pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
<category>
  <pattern>_ SCHOOL</pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
<category>
  <pattern>SCHOOL *</pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
<category>
  <pattern>_ SCHOOL *</pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>

```

Example

Update `srai.aiml` inside **C > ab > bots > test > aiml** and `srai.aiml.csv` inside **C > ab > bots > test > aimlif** directories.

srai.aiml

```

<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0.1" encoding="UTF-8"?>
  <category>
    <pattern> WHO IS ALBERT EINSTEIN </pattern>
    <template>Albert Einstein was a german physicist.</template>
  </category>
  <category>
    <pattern> WHO IS ISSAC NEWTON </pattern>
    <template>Issac Newton was a english physicist and mathematician.</template>
  </category>
  <category>
    <pattern>DO YOU KNOW WHO * IS</pattern>
    <template>
      <srai>WHO IS <star/></srai>
    </template>
  </category>
  <category>
    <pattern>BYE</pattern>
    <template>Good Bye!</template>
  </category>
  <category>
    <pattern>BYE *</pattern>
    <template>
      <srai>BYE</srai>
    </template>
  </category>
  <category>
    <pattern>FACTORY</pattern>
    <template>Development Center!</template>
  </category>
  <category>
    <pattern>INDUSTRY</pattern>
    <template>
      <srai>FACTORY</srai>
    </template>
  </category>

```

```

<category>
  <pattern>SCHOOL</pattern>
  <template>School is an important institution in a child's life.</template>
</category>
<category>
  <pattern>_ SCHOOL</pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
<category>
  <pattern>_ SCHOOL</pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
<category>
  <pattern>SCHOOL * </pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
<category>
  <pattern>_ SCHOOL * </pattern>
  <template>
    <srai>SCHOOL</srai>
  </template>
</category>
</aiml>

```

star.aiml.csv

```

0,WHO IS ALBERT EINSTEIN,* ,*,Albert Einstein was a german physicist.,srai.aiml
0,WHO IS ISSAC NEWTON,* ,*,Issac Newton was a english physicist and
mathematician.,srai.aiml
0,DO YOU KNOW WHO * IS,* ,*,<srai>WHO IS <star/></srai>,srai.aiml
0,BYE,* ,*,Good Bye!,srai.aiml
0,BYE * ,*,*,<srai>BYE</srai>,srai.aiml
0,FACTORY,* ,*,Development Center!,srai.aiml
0,INDUSTRY,* ,*,<srai>FACTORY</srai>,srai.aiml
0,SCHOOL,* ,*,School is an important institution in a child's life.,srai.aiml
0,_ SCHOOL,* ,*,<srai>SCHOOL</srai>,srai.aiml
0,SCHOOL * ,*,*,<srai>SCHOOL</srai>,srai.aiml
0,_ SCHOOL * ,*,*,<srai>SCHOOL</srai>,srai.aiml

```

Execute Program

Open command prompt. Go to **C > ab >** and type the following command:

```
java -cp lib/Ab.jar Main bot=test action=chat trace=false
```

Verify Result

```

Human: I love going to school daily.
Robot: School is an important institution in a child's life.
Human: I like my school.
Robot: School is an important institution in a child's life.
Loading [MathJax]/jax/output/HTML-CSS/jax.js

```