

# AI

BeF <bef@erlangen.ccc.de>

August 5, 2004

## artificial intelligence markup language

- intro - dialog based ai
- intro - aiml
- software requirements
- aiml tags
- aiml in use
- further reading

## natural language parsing

- intro – nlp
- first attempt – random
- analyzing text – next word method
- not intelligent language parsing
- extending the next word method





```
<BeF> i can not find my favorite book  
<maedchens> What is the title of the book?  
<BeF> harry potter  
<maedchens> /amazon harry potter...  
results: 452  
...
```

- ▶ language parsing/recognition
- ▶ answer guessing

- ▶ language parsing/recognition
- ▶ answer guessing
- ▶ learning?

## Wissensdarstellung in <XML>

## software

e.g. PyAIML, java...



```
import aiml
k = aiml.Kernel()
k.learn("std-startup.xml")
k.respond("load_aiml_b")
while True: print k.respond(raw_input(">_"))
```

## categories, patterns, templates, ...

```
<category>  
<pattern>HALLO</pattern>  
<template>Hallo!</template>  
</category>
```

## categories, patterns, templates, ...

```
<category>  
<pattern>LABER LABER LAMENTIER</pattern>  
<template>geh sterben!</template>  
</category>
```

## wildcards

```
<category>  
<pattern>* LAMINIEREN</pattern>  
<template>  
  Bitte keinen Kaese laminieren.  
</template>  
</category>
```

## wildcards

```
<category>  
<pattern>HALLO _<pattern>  
<template>Hallo.</template>  
</category>
```

```
<category>  
<pattern>* WELT<pattern>  
<template>Die Welt ist rund.</template>  
</category>
```

## wildcards

```
<category>  
<pattern>* FICKEN _<pattern>  
<template>zu mir oder zu dir?</template>  
</category>
```

## maedchens statistik...

```
$ grep -i ficken jab*log |wc -l  
58
```

## star

```
<category>  
<pattern>MEIN NAME IST _<pattern>  
<template>Hallo <star/></template>  
</category>
```



## random

```
<category>  
<pattern>* HANDKETTENSAEGE _<pattern>  
<template>  
<random>  
<li>  
    du bist eigentlich der ideale terrorist</li>  
<li>  
    Du bist bestimmt jung und student an  
    ner deutschen uni</li>  
</random>  
</template>  
</category>
```

## srai

simple redAlrection

```
<category>  
<pattern>* ODER SO<pattern>  
<template><srai><star /></srai>  
</template></category>
```

```
<category>  
<pattern>* ODER SO<pattern>  
<template><sr />  
</template></category>
```

that

```
<category><pattern>WHAT IS FNORD</pattern>  
<template><random>  
<li>Fnord is evaporated herbal tea without  
the herbs.</li>  
<li>...</li>  
</random></template></category>
```

```
<category><pattern>WHAT ELSE</pattern>  
<that>FNORD _</that>  
<template><srai>WHAT IS FNORD</srai>  
</template></category>
```

## think, get, set

```
<category><pattern>* FEFE</pattern>  
<template>Kondome schuetzen vor dem Forken.  
<think><set name="name">fefe</set></think>  
</template></category>
```



```
<BeF> i can not find my favorite book  
<maedchens> What is the title of the book?  
<BeF> harry potter  
<maedchens> /amazon harry potter...  
results: 452  
...
```

<http://www.alicebot.org/documentation/>  
<http://www.alicebot.org/TR/2001/WD-aiml/>

## simple nlp

learning by talking

## attempt 1 – random

random /usr/dict/words output...  
monkeys on the moon



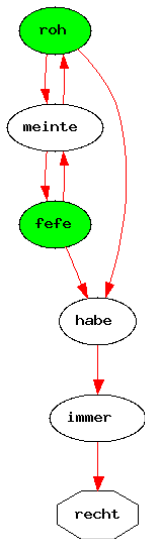
## attempt 2 – next word method

### intro

word1 word2 ... – > tree

## simple example

roh meinte fefe habe immer recht.  
fefe meinte roh habe immer recht.



## sentence construction

```
#!/usr/bin/env python
# -*- coding: iso8859-15 -*-

from aill import parser
import sys

text = file(sys.argv[1]).read()

p = parser()
p.parse(text)

for i in range(10):
    print p.sentence()
```

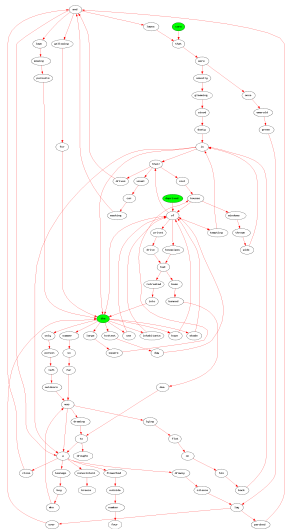
## sentence construction

test

```
fefe meinte roh habe immer recht.  
roh meinte fefe habe immer recht.  
roh habe immer recht.  
roh meinte roh habe immer recht.  
fefe meinte roh meinte fefe meinte roh habe immer recht.  
roh meinte roh meinte fefe meinte roh habe immer recht.  
fefe habe immer recht.  
fefe meinte fefe habe immer recht.  
fefe habe immer recht.  
fefe habe immer recht.
```

## another simple example

The hottest day of the summer so far was drawing to a close and a drowsy silence lay over the large, square houses of Privet Drive. Cars that were usually gleaming stood dusty in their drives and lawns that were once emerald green lay parched and yellowing -for the use of hosepipes had been banned due to drought. Deprived of their usual car-washing and lawn-mowing pursuits, the inhabitants of Privet Drive had retreated into the shade of their cool houses, windows thrown wide in the hope of tempting in a nonexistent breeze. The only person left outdoors was a teenage boy who was lying flat on his back in a flowerbed outside number four.



## another simple example

### sentence construction

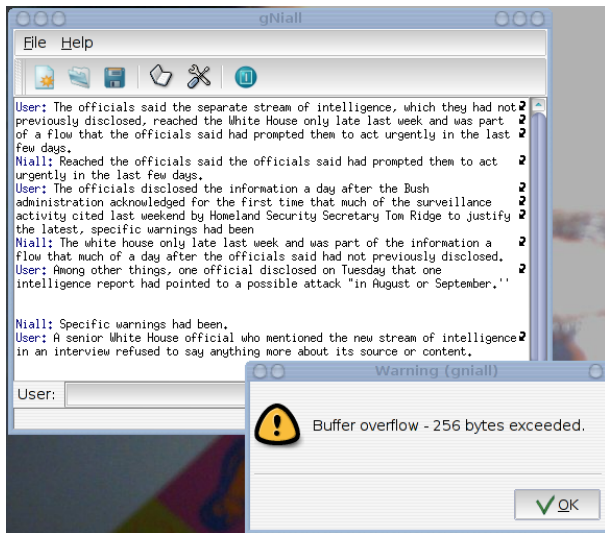
cars that were usually gleaming stood dusty in a flowerbed outside number four.

cars that were once emerald green lay parched and a close and lawn mowing pursuits the use of tempting in the summer so far was lying flat on his back in the summer so far was a drowsy silence lay parched and a teenage boy who was a nonexistent breeze.

cars that were usually gleaming stood dusty in the inhabitants of hosepipes had been banned due to drought. deprived of their usual car washing and a flowerbed outside number four.



```
Rem *****  
Rem **      NIALL  Non-Intelligent AMOS Language Learner Vers  
Rem **      =====  
Rem **                               Written By Matthew Peck in 1990  
Rem *****
```



- ▶ lookahead
- ▶ probabilities

:)

