





We can get bytes off disk and turn them into characters.

Now to find things we're interested in, inside those sequences of characters.

This week, we'll see how far we can get with rule-based systems, before we turn to machine learning approaches.





February 2011

LITERARY CHARACTER APB *for* \$400:

*His victims include Charity Burbage, Mad Eye Moody & Severus Snape; he'd be easier to catch if you'd just name him!*

BEATLES PEOPLE *for* \$200:

*“And anytime you feel the pain, hey” this guy*

*“refrain, don’t carry the world upon your shoulders”*

OLYMPIC ODDITIES *for* \$800:

*In the 2004 opening ceremonies, a sole member of this team opened the Parade of Nations; the rest of his team closed it.*

Some of these questions are really hard for people because you need to know so much trivia, and our meat minds are bad at that.

But computers can store a lot; Watson had 21.6 TB of storage, back in 2011.

LINUS'S LAW: "Given enough eyeballs, all bugs are shallow."

“Given enough text, all questions are easy.”

# Regular expressions in Python

RE functions to know:

`re.search`

`re.match`

`re.finditer`

`re.findall`

`re.compile`

`re.sub`

```
import re
```

```
s = "Hello there"
```

```
m = re.search(r"\b(t?here)\b", s)
```

```
print(m.group(1))
```

```
import re
```

```
s = "Hello there"
```

```
m = re.search(r"\b(t?here)\b", s)
```

```
print(m.group(1))
```

*Says this is a regular expression. Otherwise, \b won't be interpreted correctly as the class of word-boundary markers.*

```
import re
```

```
s = "Hello there"
```

```
m = re.match(r"\b(t?here)\b", s)
```

```
print(m.group(1))
```

```
import re  
  
s = "Hello there"  
  
m = re.match(r"\b(t?here)\b", s)  
  
print(m.group(1))
```

*Error! re.match only matches from the beginning of the string. It's equivalent to starting the RE with ^.*

→ Notebook: Regular expressions

A formal notation in which to represent the decomposition template is:

(0 YOU 0 ME)

and the reassembly rule

(WHAT MAKES YOU THINK I 3 YOU).

The “0” in the decomposition rule stands for “an indefinite number of words” (analogous to the indefinite dollar sign of COMIT) [6] while the “3” in the reassembly rule indicates that the third component of the subject decomposition is to be inserted in its place.

Weizenbaum, 1966. “*ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine*”

A formal notation in which to represent the decomposition template is:

(0 YOU 0 ME)

and the reassembly rule

(WHAT MAKES YOU THINK I 3 YOU).

```
import re



patient

 = "I hate that you judge me so much."

m = re.match("(.*)(you)(.*) (me)", patient)

print("What makes you think I " + m.group(3)
      + " you?")
```



main

eliza.py / eliza.py

↑ Top

Code

Blame

Raw



```
87     gPats = [  
92  
93         [r'Why don\'t you ([^\?]*)\??',  
94         [ "Do you really think I don't %1?",  
95           "Perhaps eventually I will %1.",  
96           "Do you really want me to %1?"]],  
97  
98         [r'Why can\'t I ([^\?]*)\??',  
99         [ "Do you think you should be able to %1?",  
100          "If you could %1, what would you do?",  
101          "I don't know -- why can't you %1?",  
102          "Have you really tried?"]],  
103  
104         [r'I can\'t (.*)',  
105         [ "How do you know you can't %1?",  
106          "Perhaps you could %1 if you tried.",  
107          "What would it take for you to %1?"]],  
108  
109         [r'I am (.*)',  
110         [ "Did you come to me because you are %1?",  
111          "How long have you been %1?",  
112          "How do you feel about being %1?"]],  
113  
114         [r'I\'m (.*)',  
115         [ "How does being %1 make you feel?",  
116          "Do you enjoy being %1?",  
117          "Why do you tell me you're %1?",  
118          "Why do you think you're %1?"]],  
119  
120         [r'Are you ([^\?]*)\??',  
121         [ "Why does it matter whether I am %1?",  
122          "Would you prefer it if I were not %1?",  
123          "Perhaps you believe I am %1.",
```

*Practice:* Information extraction

Output will be triples (entity<sub>1</sub>, relation, entity<sub>2</sub>), e.g.,

("Vassar College", "located in", "Poughkeepsie, NY")

("Grace Hopper", "born in", "1906")

“[Regular expressions] are particularly useful for searching in texts, when we have a *pattern* to search for and a *corpus* of texts to search through. A regular expression search function will search through the corpus, returning all texts that match the pattern.”

Jurafsky & Martin, § 2.1

Consider learning when people were born.

Consider learning when people were born.

What do you search for?

Let's try some of these out!

As our corpus, we'll look at part of an old snapshot of English Wikipedia...

**born in [0-9]{4}**

Don't want to match places or other descriptions,  
e.g.,

*born in New York*

*born in poverty*

→ Notebook: Information extraction

☰

David Bowie

🌐 115 languages

Article

Talk

Read

View source

View history

Tools

From Wikipedia, the free encyclopedia

★

🔒

*For other uses, see [David Bowie \(disambiguation\)](#).*

**David Robert Jones** (8 January 1947 – 10 January 2016), known as **David Bowie**,<sup>[a]</sup> was an English singer, songwriter and actor. Regarded as among the most influential musicians of the 20th century, Bowie received particular acclaim for his work in the 1970s. His career was marked by reinvention and visual presentation, and his music and stagecraft have had a great impact on [popular music](#).

Bowie studied art, music and design before embarking on a professional music career in 1963. He released a string of unsuccessful singles with local bands and [a self-titled solo album](#) (1967) before achieving his first top-five entry on the [UK singles chart](#) with "Space Oddity" (1969). After a period of experimentation, he re-emerged in 1972 during the [glam rock](#) era with the alter ego [Ziggy Stardust](#). The single "[Starman](#)" and its album *[The Rise and Fall of Ziggy Stardust and the Spiders from Mars](#)* (1972) won him widespread popularity. In 1975, Bowie's style shifted towards a sound he characterised as "[plastic soul](#)", initially alienating many of his UK fans but garnering his first major US crossover success with the number-one single "[Fame](#)" and the album *[Young Americans](#)* (1975). In 1976, Bowie starred in the cult film *[The Man Who Fell to Earth](#)* and released *[Station to Station](#)*. In 1977, he again changed

David Bowie



Bowie in 2002

Born

David Robert Jones

8 January 1947

These kind of searches let us learn lots of information that's stated in text.

Which companies bought which other companies.

What state is a town in?

Which musicians made which albums?

As we work on this information retrieval program, we've been trying to fix two kinds of errors:

- 1 ***False positives***: Matching strings that we shouldn't have matched (e.g., *born in humble circumstances*)
- 2 ***False negatives***: Not matching things that we should have matched (e.g., *born on the first of January, 1901*)

# Error types

Program thinks it  
says when  
someone was born

Program thinks it  
*doesn't* say when  
someone was born

Actually says when  
someone was born

*True positive*



*False negative*



*Doesn't* actually says  
when someone was born

*False positive*



*True negative*



In NLP, we're always dealing with these kinds of errors.

Reducing the error rate for an application often involves two antagonistic efforts:

- Increasing accuracy or precision (minimizing false positives)

- Increasing coverage or recall (minimizing false negatives)



