Computational Linguistics Introduction

Computational Linguistics: Jordan Boyd-Graber
University of Maryland

WARMUP
Roadmap

My the end of this class you should . . .

- Write regular expressions to find (linguistic) patterns
- Do simple counting using nltk
- Play around with Python interpreter
- Access corpora from nltk
- Manipulate conditional probabilities
eliza: a cautionary tale

- Claim: an electronic psychiatrist
- Is there anything interesting going on?

http://www.masswerk.at/elizabot/
What are eliza’s tricks?

<table>
<thead>
<tr>
<th>I feel Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>How often do you feel Y?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I want Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppose you got Y soon …</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>If Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you think that it’s likely that Y?</td>
</tr>
</tbody>
</table>

Other tricks
- Convert “my” to “your” in reply (and other pronouns)
- Randomly produce a change of subject if no rule matches: “tell me about your mother”
How do they do it?

- eliza is about finding patterns
- But users can type many different things
- We thus need a system for expressing many general patterns
How do they do it?

- eliza is about finding patterns
- But users can type many different things
- We thus need a system for expressing many general patterns
- Regular expressions
Wait a minute!

- Very stupid
- Brute-force
Wait a minute!

- Very elegant
- Low resource
Wait a minute!

- Very elegant
- Low resource
- But still require clever humans to write
Wait a minute!

- Very elegant
- Low resource
- But still require clever humans to write
- Even if you know regexps inside and out, it’s important know how to apply them to language
Why in an NLP course?

- Searching for linguistic phenomena (does eat ever take the object “loss”)?
- Creating features for supervised algorithms
- Useful for morphology
- Thinking about regular expressions (nice tool) will help you think about finite state machines (theoretical framework)
## Symbols and Operators

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>[]</td>
<td>Set of characters</td>
</tr>
<tr>
<td>^</td>
<td>Start of line / Negation</td>
</tr>
<tr>
<td>$</td>
<td>End of the line</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Range of Characters</td>
</tr>
<tr>
<td>+</td>
<td>At least one appearance</td>
</tr>
<tr>
<td>*</td>
<td>Any number of appearances</td>
</tr>
<tr>
<td>{N}</td>
<td>Exactly (N) appearances</td>
</tr>
</tbody>
</table>
Regular Expression Syntax

Sets

\d \text{ digits}
\D \text{ non-digits}
\s \text{ whitespace}
\S \text{ non-whitespace}
\w \text{ “words”}
\W \text{ non-“words”}
\b \text{ empty string at word start}
. \text{ any character except for newline}
Sets

\d \text{ digits} \quad [0-9]
\D \text{ non-digits} \quad [^0-9]
\s \text{ whitespace} \quad [\t\n\r\f\v]
\S \text{ non-whitespace} \quad [^\t\n\r\f\v]
\w \text{ “words”} \quad [a-zA-Z0-9_\ ]
\W \text{ non-“words”} \quad [^a-zA-Z0-9_\ ]
\b \text{ empty string at word start} \quad \W\b\W
. \text{ any character except for newline} \quad b.d
Backreference

- If you enclose a subexpression in parens \( a . \)
- You can reference that expression again \( \backslash 1 \) (for most recent)
- For less recent, the numbers increment \( \backslash 2 \), etc.
Ranges

What does this RegEx do?
\b[a-z]+l
Examples

Ranges

What does this RegEx do?

\b[a-z]+l

```
^I\b.*$  

I am the very model of a modern Major-General, 
I've information vegetable, animal, and mineral, 
I know the kings of England, and I quote the fights historical
From Marathon to Waterloo, in order categorical; a
I'm very well acquainted, too, with matters mathematical, 
I understand equations, both the simple and quadratical, 
About binomial theorem I'm teeming with a lot o' news, (bothered for a rhyme)
With many cheerful facts about the square of the hypotenuse.
```
Ranges

What does this RegEx do?

[aeiou]{2,}
Ranges

What does this RegEx do?

[aeiou]{2,}

I am I the very model of a modern Major-General,
I've information vegetable, animal, and mineral,
I know the kings of England, and I quote the fights historical
From Marathon to Waterloo, in order categorical;
I'm very well acquainted, too, with matters mathematical,
I understand equations, both the simple and quadratical,
About binomial theorem I'm teeming with a lot o' news, (bothered for a rhyme)
With many cheerful facts about the square of the hypotenuse.
Ranges

What does this RegEx do?

[^aeiou]{2,}
Ranges

What does this RegEx do?

[^aeiou]{2,}

I am I the very model of a modern Major-General,
I've information vegetable, animal, and mineral,
I know the kings of England, and I quote the fights historical
From Marathon to Waterloo, in order categorical;
I'm very well acquainted, too, with matters mathematical,
I understand equations, both the simple and quadratical,
About binomial theorem I'm teeming with a lot o' news, (bothered for a rhyme)
With many cheerful facts about the square of the hypotenuse.
Ranges

What does this RegEx do?

[^aeiou\W]{2,}

I am I the very model of a modern Major-General,
I've information vegetable, animal, and mineral,
I know the kings of England, and I quote the fights historical
From Marathon to Waterloo, in order categorical;
I'm very well acquainted, too, with matters mathematical,
I understand equations, both the simple and quadratical,
About binomial theorem I'm teeming with a lot o' news, (bothered for a rhyme)
With many cheerful facts about the square of the hypotenuse.
Backreference

What does this RegEx do?

\b\w*\1\w*\b
Backreference

What does this RegEx do?
\b\w*(.)\1\w*\b

I am I the very model of a modern Major-General,
I've information vegetable, animal, and mineral,
I know the kings of England, and I quote the fights historical
From Marathon to *Waterloo*, in order categorical;
I'm very well acquainted, too, with matters mathematical,
I understand equations, both the simple and quadratical,
About binomial theorem I'm *teeming* with a lot o' news, (bothered for a rhyme)
With many *cheerful* facts about the square of the hypotenuse.
Exercises

Thou Must

Challenge

Find all examples of “thou ___t” in the bible; what are the most frequent?

- nltk.corpus.gutenberg
- import re
- FreqDist or Counter
Exercises

Thou Must
Exercises

Thou Must

```python
thou_regexp = re.compile(r"[Tt]hou\s[\w]*t\s")
thou_count = FreqDist()
for ii in thou_regexp.findall(gutenberg.raw('bible-kjv.txt')):
    thou_count[ii] += 1
thou_count.tabulate(5)
```
Exercises

Find a Street

Challenge

Find all examples of “Capital Word” Street in all of the Gutenberg text.
Exercises

Find a Street

street_regexp = re.compile(r"[A-Z]\w* \[S\]treet")
Find a Street

```python
street_regexp = re.compile(r"[A-Z]\w*\s[S]treet")
for fileid in gutenberg.fileids():
    print(fileid, street_regexp.findall(gutenberg.raw(fileid)))
```
Repeated Words

Challenge

1. Find all examples of repeated words in all of Gutenberg.
2. Find all examples of repeated words separated by some other word in Gutenberg.

- finditer
- group
- Back references
Repeated Words
Repeated Words

```python
repeat_regexp = re.compile(r'\b(\w+)\s(\1\b)+')
for fileid in gutenberg.fileids():
    matches = list(repeat_regexp.finditer(gutenberg.raw(fileid)))
    print(fileid, [x.group(0) for x in matches])
```
Exercises

Repeated Words (with something in between)
Repeated Words (with something in between)

repeat_regexp = re.compile(r"\b(\w+)\s\w+\s(\1\b)+")
for fileid in gutenberg.fileids():
    matches = list(repeat_regexp.finditer(gutenberg.raw(fileid)));
    print(fileid, [x.group(0) for x in matches])
Regexp Golf

REGEX GOLF:
YOU TRY TO MATCH ONE
GROUP BUT NOT THE OTHER.
/M | [TN]B/ MATCHES
STAR WARS SUBTITLES
BUT NOT STAR TREK.
COOL.
### Regexp Golf

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>afoot</td>
<td></td>
<td>Atlas</td>
</tr>
<tr>
<td>tick</td>
<td></td>
<td>trickingly</td>
</tr>
<tr>
<td>abac</td>
<td></td>
<td>beam</td>
</tr>
<tr>
<td>undergrounder</td>
<td></td>
<td>hypergoddess</td>
</tr>
<tr>
<td>civic</td>
<td></td>
<td>cinnabar</td>
</tr>
<tr>
<td>unintelligibility</td>
<td></td>
<td>unregainable</td>
</tr>
</tbody>
</table>
## Regexp Golf

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>foo</code></td>
<td>afoot</td>
<td>Atlas</td>
</tr>
<tr>
<td></td>
<td>tick</td>
<td>trickingly</td>
</tr>
<tr>
<td></td>
<td>abac</td>
<td>beam</td>
</tr>
<tr>
<td></td>
<td>undergrounder</td>
<td>hypergoddess</td>
</tr>
<tr>
<td></td>
<td>civic</td>
<td>cinnabar</td>
</tr>
<tr>
<td></td>
<td>unintelligibility</td>
<td>unregainable</td>
</tr>
</tbody>
</table>
## Regexp Golf

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>afoot</td>
<td>Atlas</td>
</tr>
<tr>
<td>k$</td>
<td>tick</td>
<td>trickingly</td>
</tr>
<tr>
<td></td>
<td>abac</td>
<td>beam</td>
</tr>
<tr>
<td></td>
<td>undergradner</td>
<td>hypergoddess</td>
</tr>
<tr>
<td></td>
<td>civic</td>
<td>cinnabar</td>
</tr>
<tr>
<td></td>
<td>unintelligibility</td>
<td>unregainable</td>
</tr>
</tbody>
</table>
# Regexp Golf

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>afoot</td>
<td>Atlas</td>
</tr>
<tr>
<td>k$</td>
<td>tick</td>
<td>trickingly</td>
</tr>
<tr>
<td>^[a-f]+$</td>
<td>abac</td>
<td>beam</td>
</tr>
<tr>
<td></td>
<td>undergrounder</td>
<td>hypergoddess</td>
</tr>
<tr>
<td></td>
<td>civic</td>
<td>cinnabar</td>
</tr>
<tr>
<td></td>
<td>unintelligibility</td>
<td>unregainable</td>
</tr>
</tbody>
</table>
## Regexp Golf

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>foo</code></td>
<td><code>afoot</code></td>
<td><code>Atlas</code></td>
</tr>
<tr>
<td><code>k$</code></td>
<td><code>tick</code></td>
<td><code>trickingly</code></td>
</tr>
<tr>
<td><code>^[a-f]+$</code></td>
<td><code>abac</code></td>
<td><code>beam</code></td>
</tr>
<tr>
<td><code>\w3.*\1</code></td>
<td><code>undergrounder</code></td>
<td><code>hypergoddess</code></td>
</tr>
<tr>
<td></td>
<td><code>civic</code></td>
<td><code>cinnabar</code></td>
</tr>
<tr>
<td></td>
<td><code>unintelligibility</code></td>
<td><code>unregainable</code></td>
</tr>
</tbody>
</table>
**Regexp Golf**

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>afoot</td>
<td>Atlas</td>
</tr>
<tr>
<td>k$</td>
<td>tick</td>
<td>trickingly</td>
</tr>
<tr>
<td>^[a-f]+$</td>
<td>abac</td>
<td>beam</td>
</tr>
<tr>
<td>(\w3).*\1</td>
<td>undergrounder</td>
<td>hypergoddess</td>
</tr>
<tr>
<td>(.)(.)..?\2\1</td>
<td>civic</td>
<td>cinnabar</td>
</tr>
<tr>
<td></td>
<td>unintelligibility</td>
<td>unregainable</td>
</tr>
</tbody>
</table>
### Regexp Golf

<table>
<thead>
<tr>
<th>Regexp</th>
<th>Matches</th>
<th>Doesn’t Match</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>foo</code></td>
<td>afoot</td>
<td>Atlas</td>
</tr>
<tr>
<td><code>k$</code></td>
<td>tick</td>
<td>trickingly</td>
</tr>
<tr>
<td><code>^[a-f]+$</code></td>
<td>abac</td>
<td>beam</td>
</tr>
<tr>
<td><code>(\w3).*\1</code></td>
<td>undergrounder</td>
<td>hypergoddess</td>
</tr>
<tr>
<td><code>(\.)\.(\.)\.?\2\1</code></td>
<td>civic</td>
<td>cinnabar</td>
</tr>
<tr>
<td><code>(\.)\.(\.)\{3\}</code></td>
<td>unintelligibility</td>
<td>unregainable</td>
</tr>
</tbody>
</table>
Changin Gears: Bayes Rule

There's a test for Boogie Woogie Fever (BWF). The probability of getting a positive test result given that you have BWF is 0.8, and the probability of getting a positive result given that you do not have BWF is 0.01. The overall incidence of BWF is 0.01.

1. What is the marginal probability of getting a positive test result?
2. What is the probability of having BWF given that you got a positive test result?
One coin in a collection of 65 has two heads. The rest are fair. If a coin, chosen at random from the lot and then tossed, turns up heads 6 times in a row, what is the probability that it is the two-headed coin?