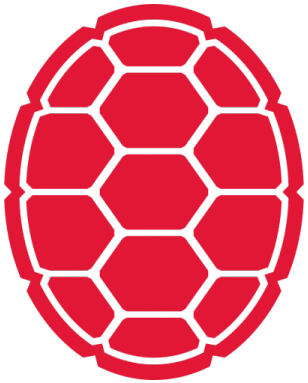


# Computational Linguistics Introduction

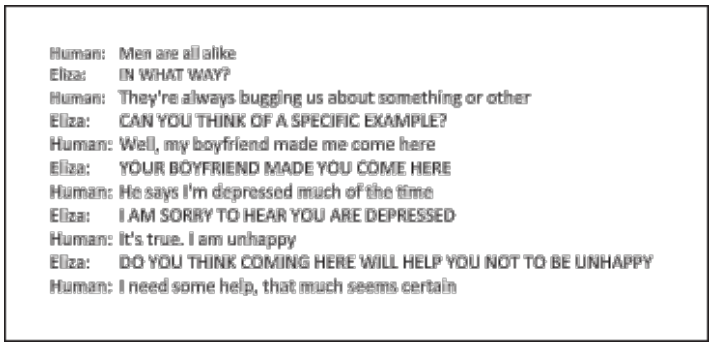
Computational Linguistics: Jordan Boyd-Graber
University of Maryland
WARMUP

## 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



Human: Men are all alike
Eliza: IN WHAT WAY?
Human: They're always bugging us about something or other
Eliza: CAN YOU THINK OF A SPECIFIC EXAMPLE?
Human: Well, my boyfriend made me come here
Eliza: YOUR BOYFRIEND MADE YOU COME HERE
Human: He says I'm depressed much of the time
Eliza: I AM SORRY TO HEAR YOU ARE DEPRESSED
Human: It's true. I am unhappy
Eliza: DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
Human: I need some help, that much seems certain

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

http://www.masswerk.at/elizabot/

## What are eliza's tricks?

### I feel *Y*

How often do you feel *Y*?

### I want *Y*

Suppose you got *Y* soon . . .

### If *Y*

Do you think that it's likely that *Y*?

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

| Symbol | Meaning |
|---|---|
| [] | Set of characters |
| ^ | Start of line / Negation |
| $ | End of the line |
| \| | Or |
| - | Range of Characters |
| + | At least one appearance |
| * | Any number of appearances |
| $\{N\}$ | Exactly $N$ appearances |

## Sets

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

## Sets

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

## Backreference

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

## Ranges

### What does this RegEx do?

\b[a-z]+l

## Ranges

### What does this RegEx do?

\b[a-z]+l

```
^I|\.$
```

```
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,}

```
[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;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,}

```
[^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;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\W]{2,}

```
[^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;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.
```

## Backreference

### What does this RegEx do?

\b\w*(.)\1\w*\b

## Backreference

### What does this RegEx do?

\b\w*(.)\1\w*\b

```
\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.
```

## 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`

# Thou Must

## Thou Must

```
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)
```

## Find a Street

### Challenge

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

# Find a Street

## Find a Street

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

## 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

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

# 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(fil
    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

| Regexp | Matches | Doesn't Match |
|---|---|---|
| | afoot | Atlas |
| | tick | trickingly |
| | abac | beam |
| | undergrounder | hypergoddess |
| | civic | cinnabar |
| | unintelligibility | unregainable |

## Regexp Golf

| Regexp | Matches | Doesn't Match |
|---|---|---|
| `foo` | afoot | Atlas |
| | tick | trickingly |
| | abac | beam |
| | undergrounder | hypergoddess |
| | civic | cinnabar |
| | unintelligibility | unregainable |

## Regexp Golf

| Regexp | Matches | Doesn't Match |
|---|---|---|
| `foo` | afoot | Atlas |
| `k$` | tick | trickingly |
| | abac | beam |
| | undergrounder | hypergoddess |
| | civic | cinnabar |
| | unintelligibility | unregainable |

## Regexp Golf

| Regexp | Matches | Doesn't Match |
|---|---|---|
| `foo` | afoot | Atlas |
| `k$` | tick | trickingly |
| `^[a-f]+$` | abac | beam |
| | undergrounder | hypergoddess |
| | civic | cinnabar |
| | unintelligibility | unregainable |

## Regexp Golf

| Regexp | Matches | Doesn't Match |
|---|---|---|
| `foo` | afoot | Atlas |
| `k$` | tick | trickingly |
| `^[a-f]+$` | abac | beam |
| `(\w3).*\1` | undergrounder | hypergoddess |
| | civic | cinnabar |
| | unintelligibility | unregainable |

## Regexp Golf

| Regexp | Matches | Doesn't Match |
|---|---|---|
| `foo` | afoot | Atlas |
| `k$` | tick | trickingly |
| `^[a-f]+$` | abac | beam |
| `(\w3).*\1` | undergrounder | hypergoddess |
| `(.)(.).?\2\1` | civic | cinnabar |
| | unintelligibility | unregainable |

## Regexp Golf

| Regexp | Matches | Doesn't Match |
|---|---|---|
| `foo` | afoot | Atlas |
| `k$` | tick | trickingly |
| `^[a-f]+$` | abac | beam |
| `(\w3).*\1` | undergrounder | hypergoddess |
| `(.)(.).?\2\1` | civic | cinnabar |
| `(.)(.\1){3}` | unintelligibility | unregainable |

## Changin Gears: Bayes Rule

There's a test for Boogie Woogie Fever (BWF). The probability of geting 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?

## Conditional Probabilities

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?