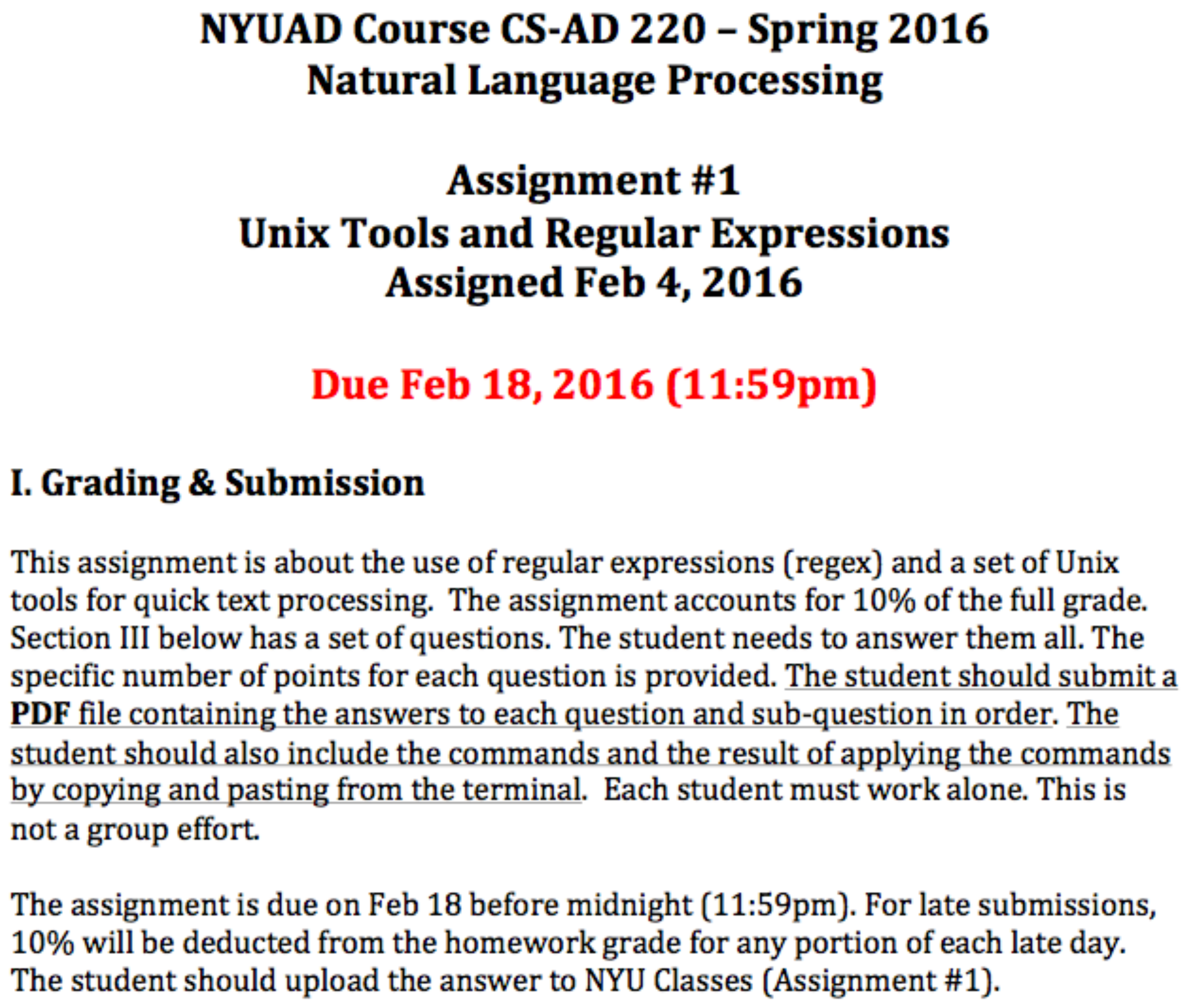
CS-AD 220 – Spring 2016

Natural Language Processing

Session 4: 9-Feb-16

Prof. Nizar Habash

Some slides are adapted from Jurafsky and Martin's course slides on Speech and Language Processing



NYUAD Course CS-AD 220 – Spring 2016
Natural Language Processing

Assignment #1
Unix Tools and Regular Expressions
Assigned Feb 4, 2016

Due Feb 18, 2016 (11:59pm)

I. Grading & Submission

This assignment is about the use of regular expressions (regex) and a set of Unix tools for quick text processing. The assignment accounts for 10% of the full grade. Section III below has a set of questions. The student needs to answer them all. The specific number of points for each question is provided. The student should submit a **PDF** file containing the answers to each question and sub-question in order. The student should also include the commands and the result of applying the commands by copying and pasting from the terminal. Each student must work alone. This is not a group effort.

The assignment is due on Feb 18 before midnight (11:59pm). For late submissions, 10% will be deducted from the homework grade for any portion of each late day. The student should upload the answer to NYU Classes (Assignment #1).

Assignment #1 posted on NYU Classes

Moving Legislative Day Class

- Spring Break is March 18 – 25, 2016
- Sat March 26, 2016 is a Legislative *Thursday*
  - Move to
    - Sat April 2, 2016 or
    - Sun April 3, 2016
  - What time?

Summer Internship in NLP at CAMeL LAB

- https://students.nyuad.nyu.edu/academics/research/research-opportunities/
  - Arabic Dialect Chatbot Project
  - Arabic Machine Translation Star Project
- Application deadline – March 1, 2016

Text Processing Review

- **Regex** **matches**
  - /an/
  - /[an]/
  - /[^an]/
  - /[A-Z]/
  - /./
  - /a+/
  - /a*/
  - /a?/

Text Processing Review

- **Regex** **matches**
  - /\s/
  - /\S/
  - /\w/
  - /\W/
  - /d/
  - /D/
  - /\n/

Text Processing Review

- **Regex** **matches**

  Anchors

  – /^/
  – /$/
  – /\b/
  – /\B/

  /^[man]/
  /^[man]$/
  /s\b/
  /s$/
  /s\B/

Text Processing Review

- **Regex matches**

  Group

  - /(ha)+/
  - /(ha|he)+/
  - Backreference
    - /(ha|he)\1/

Unix Commands

- **cat**:
- **more**:
- **head -<number>**:
- **tail -<number>**:
- **|** :
- **>** :
- **wc**:
- **sort** :
  - **sort -u** :
  - **sort -nr** :
- **uniq -c** :

Unix Commands

- **egrep**: globally search for a regular expression and print
  - egrep –i   ignore case
  - egrep –v   invert match

```
cat Bible-English.txt |wc
   31102  789635 4138512
cat Bible-English.txt |egrep '\bAnd\b' |wc
   12029  331758 1727495
cat Bible-English.txt |egrep '\band\b' |wc
   20736  576950 3026233
cat Bible-English.txt |egrep -i '\band\b' |wc
   23867  646949 3388601
cat Bible-English.txt |egrep -vi '\band\b' |wc
    7235  142686  749911
```

Substitutions

- Perl is a programing language. We will use it here only for processing strings
- tr/X/Y/ : translate command
  - Works on characters only
  - Lower case all words
    - cat Bible-English.txt|perl -pe 'tr/A-Z/a-z/;’
  - Shift vowels
    - cat Bible-English.txt |perl -pe 'tr/aeiuo/iaeou/;'|head -1
    - In tha bagenneng Gud craitad tha haivan ind tha airth.

Substitutions

- s/X/Y/g : substitute command
  - Works on strings
  - Change *art thou* to *are you*

```
cat Bible-English.txt |grep 'art thou'|head -1
And the LORD God called unto Adam, and said unto him, Where art thou?

cat Bible-English.txt |grep 'art thou'|head -1|perl -pe 's/\bart thou\b/are
you/'
And the LORD God called unto Adam, and said unto him, Where are you?
```

Substitutions

- s/X/Y/g : substitute command
  - Works on strings
  - Split sentences into one word per line

```
cat Bible-English.txt |perl -pe 's/\s/\n/g;'|more
 In
 the
 beginning
 God
 created
 the
 heaven
 and
 the
 earth.
```

Substitutions

- What are the 10 most frequent words in this text?

```
cat Bible-English.txt |perl -pe 's/\s/\n/g;'|sort |uniq -c |sort -
nr |head -10

62057 the
38579 and
34389 of
13364 to
12736 And
12453 that
12166 in
9760 shall
9510 he
8932 unto
```

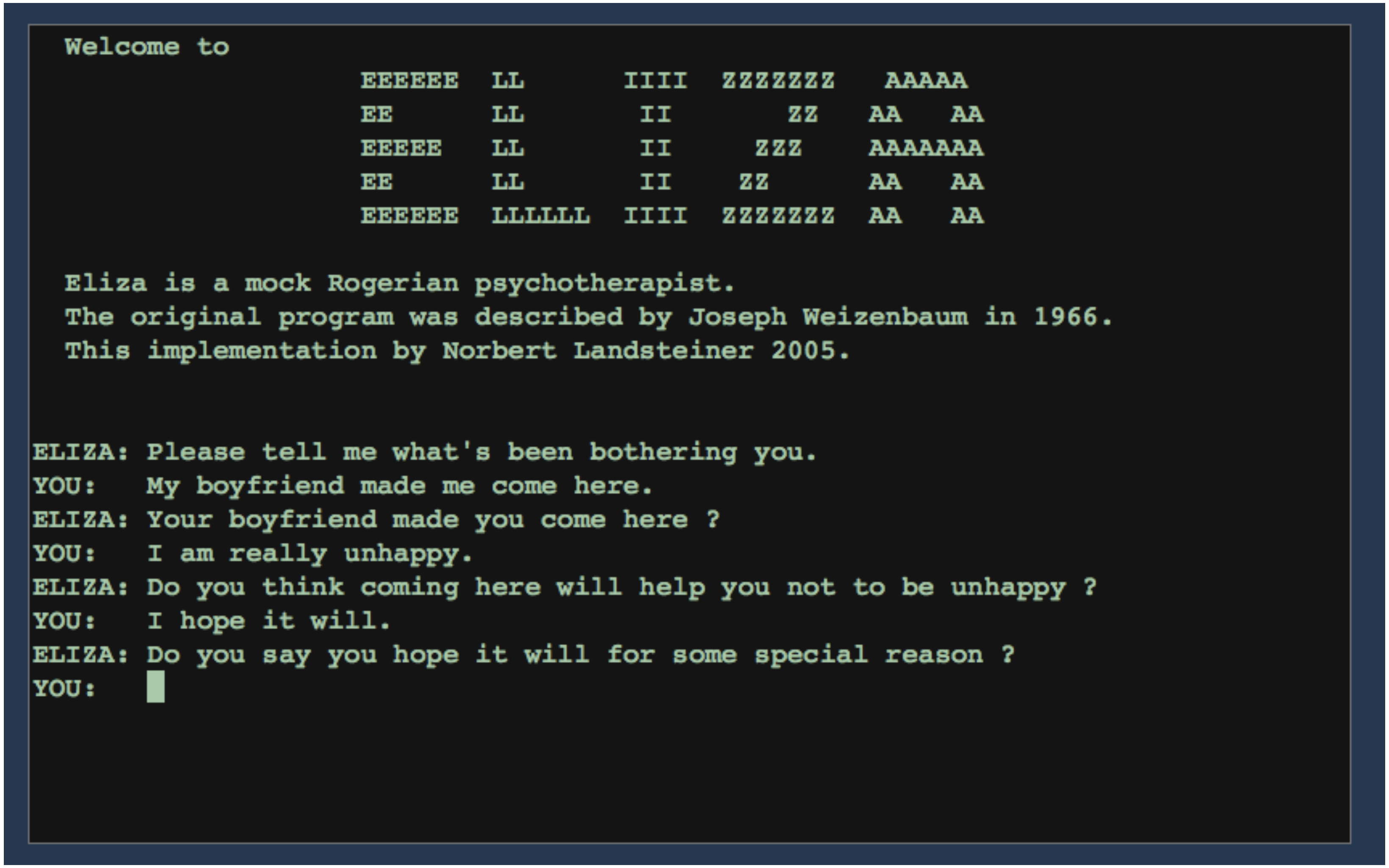
Substitution

- Memory
  - Capture with a pair of parentheses “()”
  - $1 in substitution
  - s/(ABC)/$1$1/g; changes “ABC” to “ABCABC”

  - Double all vowels

```
cat Bible-English.txt |perl -pe 's/([AEIOUaeiou])/$1$1/g;'|
head -1
IIn thee beegiinniing Good creeaateed thee heeaaveen aand thee
eeaarth.
```

Please tell me what's been bothering you.



```
  Welcome to
                    EEEEEE  LL      IIII  ZZZZZZZ   AAAAA
                    EE      LL       II       ZZ   AA   AA
                    EEEEE   LL       II     ZZZ    AAAAAAA
                    EE      LL       II    ZZ      AA   AA
                    EEEEEE  LLLLLL  IIII  ZZZZZZZ  AA   AA

  Eliza is a mock Rogerian psychotherapist.
  The original program was described by Joseph Weizenbaum in 1966.
  This implementation by Norbert Landsteiner 2005.


ELIZA: Please tell me what's been bothering you.
YOU:   My boyfriend made me come here.
ELIZA: Your boyfriend made you come here ?
YOU:   I am really unhappy.
ELIZA: Do you think coming here will help you not to be unhappy ?
YOU:   I hope it will.
ELIZA: Do you say you hope it will for some special reason ?
YOU:
```

Unix Commands

- “cat |” allows you to type the input without creating a file, e.g.

```
cat |perl -pe 'tr/A-Z/a-z/'
HI
hi
```

$\leftarrow$ ***input***

$\leftarrow$ ***output***

Unix Commands

- Make a file with a number of substitutions and call from the prompt

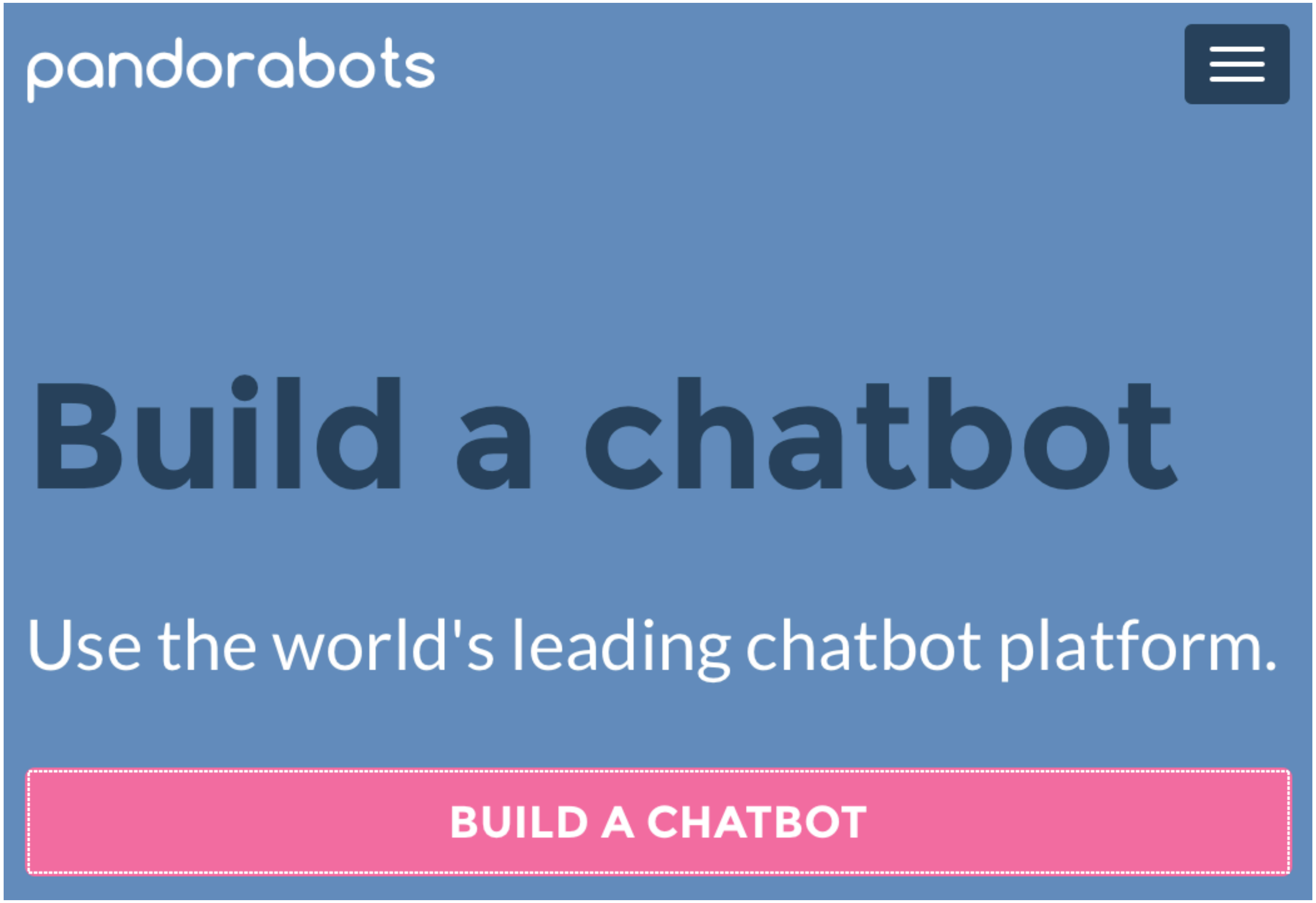
```
cat > X
tr/a-z/A-Z/;
s/(\d+)/[$1]/;
^C

cat | perl -p X
hi
HI
this is a number 12
THIS IS A NUMBER [12]
```

Simple Eliza code ***eliza.pl***

```
#Upper case all characters
tr/a-z/A-Z/;

#Remove all non-words except for spaces
s/[^\w\s]//;

#Switch first person to second person
s/\bMY\b/YOUR/g;
s/\b(I|ME)\b/YOU/g;
s/\bAM\b/ARE/g;

#Responses
s/.*\b(HI|HELLO|HOWDY)\b.*/$1 THERE. HOW ARE YOU DOING?/;
s/.*\bYOU ARE (DEPRESSED|SAD|UNHAPPY)\b.*/I AM SORRY TO HEAR YOU ARE $1./;
s/.*\bALL\b.*/IN WHAT WAY?/;
s/.*\bALWAYS\b.*/CAN YOU THINK OF A SPECIFIC EXAMPLE?/;
s/.*\bNOT\b.*/WHY NOT?/;
s/.*\bYES\b.*/I SEE/;

#Add prompt
s/^/>> /;
```

Run eliza.pl

```
cat |perl –p eliza.pl
Hi Eliza
>> HI THERE. HOW ARE YOU DOING?
I am not ok.
>> WHY NOT?
My wife always nags me
>> CAN YOU THINK OF A SPECIFIC EXAMPLE?
she made me come here today
>> SHE MADE YOU COME HERE TODAY
yes
>> I SEE
she says I am depressed
>> I AM SORRY TO HEAR YOU ARE DEPRESSED.
```



pandorabots

Build a chatbot

Use the world's leading chatbot platform.

https://playground.pandorabots.com/en/

More exercises with Regex and Unix commands

Word Unigrams

- Get the counts of the all the words in some document

```
cat Bible-English.txt |perl -pe 's/\s/\n/g;' |sort |
uniq -c |sort -nr
```

```
62057 the
38579 and
34389 of
13364 to
12736 And
12453 that
12166 in
9760 shall
9510 he
8932 unto
```

…

```
1 edification.
1 edges;
1 edges,
1 edge?
1 edge,
1 ebony.
1 eating:
1 eateth;
1 eateth:
1 eaters
1 eater:
```

Word Unigrams

- Get the counts of the all the words in some document
  - What is a word?

    MAN!!   Man!   Man   man
  - Tokens (all instances) vs Types (unique tokens)

```
cat Bible-English.txt |perl -pe 's/[^a-zA-Z]+/\n/g;'|egrep  '\w'|wc
  792074  792074 4014578

cat Bible-English.txt |perl -pe 's/\s/\n/g;' |sort -u|wc
   28881   28880  241737

cat Bible-English.txt |perl -pe 's/[^a-zA-Z]+/\n/g;'|egrep  '\w'|sort -u|wc
   13563   13563  107609

cat Bible-English.txt |perl -pe 's/[^a-zA-Z]+/\n/g; tr/A-Z/a-z/;'|egrep
'\w'|sort -u|wc
   12498   12498  100376
```

Character Unigrams

- Get the frequency of letters in a text

Character Unigrams

- Get the frequency of letters in a text

```
cat Bible-English.txt |perl -pe 'tr/A-Z/a-z/; s/[^a-zA-Z]//g; s/(\S)/$1\n/g;'|sort |uniq -c
|sort -nr
```

```
410120 e        82946 u
316030 t        79534 m
282019 h        65217 w
274641 a        58247 y
241601 o        54856 g
223953 n        54430 c
192840 i        48562 b
189142 s        42742 p
169124 r        30253 v
157553 d        22110 k
129353 l        8779 j
83092 f         2957 z
                1450 x
                 953 q
```

Four-Letter Words

- What percentage of words in the Bible consists of four-letters?

```
cat Bible-English.txt |perl -pe 's/[^a-zA-Z]+/\n/g;'| egrep
'^\w\w\w\w$'|wc
  176249  176249  881245
```

```
cat Bible-English.txt |perl -pe 's/[^a-zA-Z]+/\n/g;'| egrep
'\w'|wc
  792074  792074 4014578
```

➔ 22%

Pig Latin

- Pig Latin is a language game in which English words are altered in a systematic way producing odd sounding sentences that are incomprehensible to people who are not aware of the rules.

- Consider the following rules
  - For words that begin with a consonant letter, the initial consonant letter is moved to the end of the word, and "ay" is added:
    - Pig Latin → igpay atinlay
    - banana → ananabay
    - Spain → painsay
    - The → hetay
  - For words that begin with vowel, just add "way" to the end:
    - egg → eggway
    - inbox → inboxway
    - a → away

- Write regular expression substitution commands to translate from English to Pig Latin.

Next Time

- Finite State Automata
- Do the reading!