Python Programming

Eun Woo Kim

Big Data Camp

(May 11th, 2016)

As a beginner of programming..

- Code is confusing **V**
- Don’t know if I can do programming.. **V**
- Don’t know what I can do with Python.. **Reed**

I am here to share with you

"Six things I wish I had known a year ago

about Python Programming"

(1) Need many packages (or modules)

import XXX
import YYY
import ZZZ

package or module

import os -- operating system interface
import re -- string processing
import csv -- csv file reading/writing
import nltk -- natural language processing

import statistics

statistics.mean([1,2,3,4,5])

function / method

You may have to import many modules.

Don't worry about it.

(2) Directory matters

import os

os.getcwd() -- get current working directory

os.chdir('U:\\Big Data Camp') -- change the current working directory

os.listdir() -- returns a list of sub directories and file in this path

os.mkdir('folder1') -- make a new directory

os.rename('folder1', 'newfolder') -- renaming a directory

os.rename('test1.txt', 'newname.txt') -- renaming a file

(3) Reading/writing a file needs a practice

A. Reading a file

open('name1.txt')

list(open('name1.txt'))

import csv

with open('name1.txt', 'r') as f:

```
with open('name1.txt', 'r') as f:
    csv_read = csv.reader(f, delimiter='\t')
    for a in csv_read:
        print(a[0:3])
```

word1	word2	word3

line1

line2

line3

['word1\tword2\tword3']

['line1\n', 'line2\n', 'line3']

['word1', 'word2', 'word3']

['line1', 'line2', 'line3']

(3) Reading/writing a file needs a practice

word1	word2	word3

['word1\tword2\tword3']

hello

B. Writing a file

```
open('name1.txt')
list(open('name1.txt'))
```

```
with open('name1.txt', 'w') as g:
    g.write('hello')
```

(4) Always write comments

```
# specify how many tweets I want
totalNumTweet = 10000

def writeResult (scores):
# example scores entry:
#  {'1_U of M' : {'innovation': {2015: 92, 2016: 93},
#                 'donation': {2015: 85, 2016: 90} } }
```

Comments help you remember what your code is for.

Comments help you think clearly.

(5) Googling is ok, actually very common and recommended

- Try running your code as you write.

  - when you encounter an error,

    think about what could have been the problem.

  - if you cannot figure out the problem by yourself, google!

- Online resources: Python tutorial, Stackoverflow

  -There can be multiple answers to one question.

  -It is still hard to figure out which answer is the best.

  -Start with one answer that seems reasonable and which you can understand the most.

(6) It is like learning a foreign language

- It takes a long time
- You need to learn grammars, vocabularies, sentence structures, etc.
- There are many ways of writing codes
- Compare your codes with other people’s codes
- You have to practice a lot (trial and error)
- Talk with other people who use Python or who do programming
- Think about why you want to learn Python
- If you like it, you learn fast

What I did after Big Data Camp

(1) Took class: Ling 441 ‘Computational Linguistics’

(2) Tried using Python instead of Excel!

(3) Used Python and API for my research project

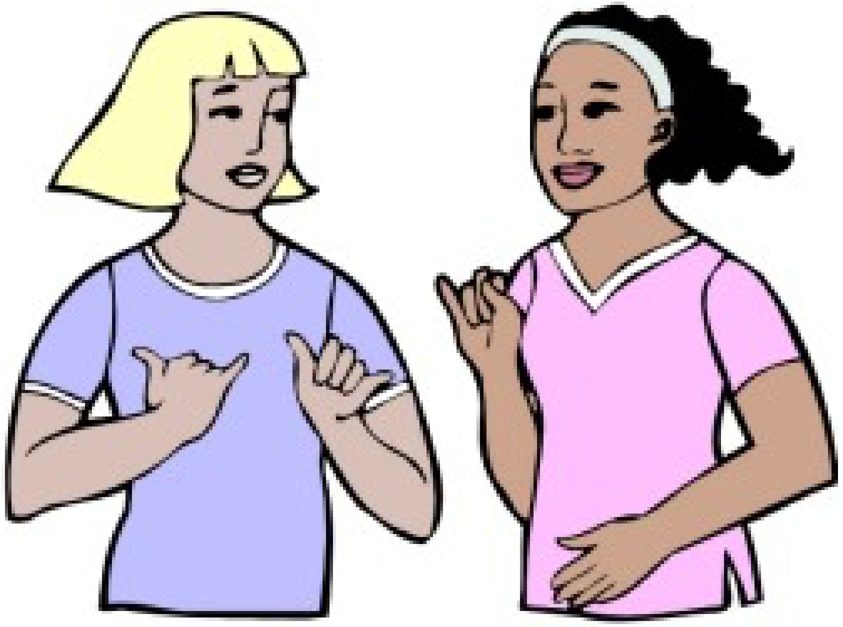
```
print('T'+'H'+'A'+'N'+'K'+' '+'Y'+'O'+'U'+'!')
```

Natural Language Processing for Understanding Big Data

Reed Coke

What is Natural Language Processing (NLP)?

- Humans interact with each other using spoken, written, or signed **natural language**.



What is Natural Language Processing (NLP)?

- Humans interact with each other using spoken and/or written **natural language**.
- Computers interact with each other (ultimately) using **binary**.

10101110101000101010101010

What is Natural Language Processing (NLP)?

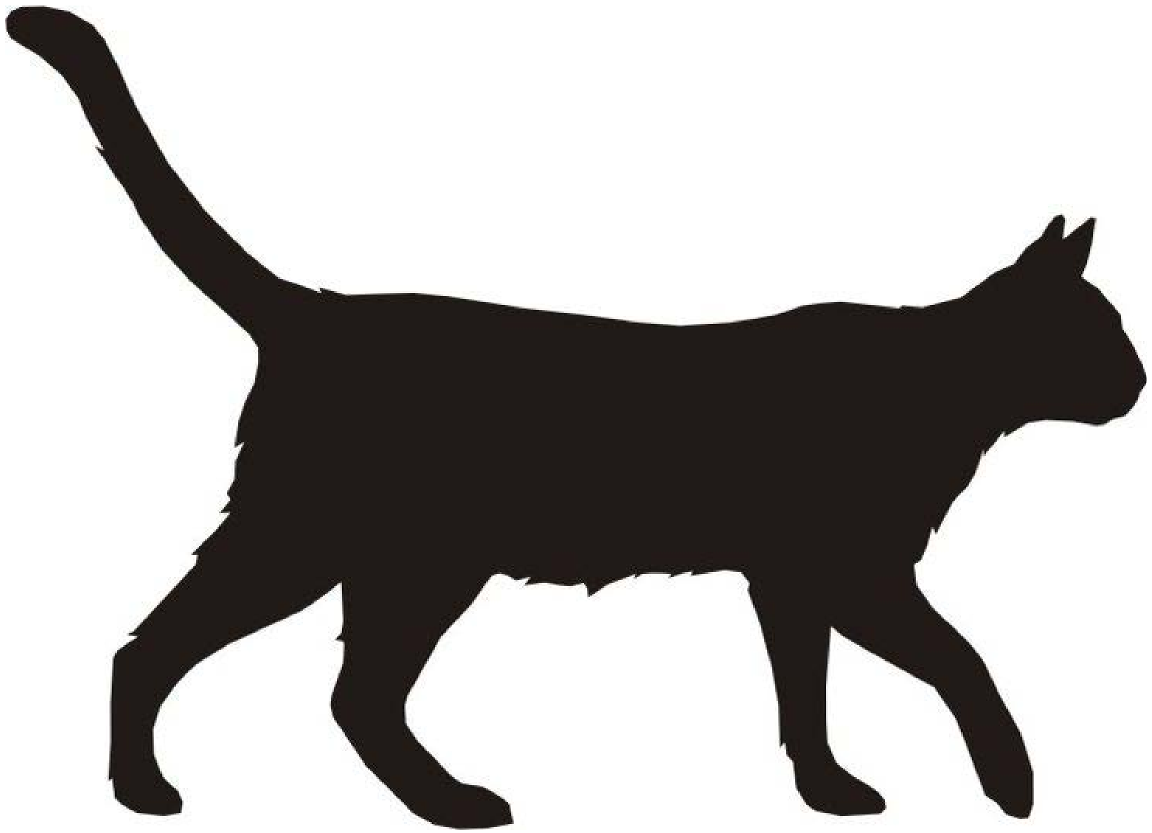
- Humans interact with each other using spoken and/or written **natural language**.
- Computers interact with each other (ultimately) using **binary**.
- NLP is concerned with getting computers to translate from natural language to binary and back.

Outline

- Why is NLP hard?
- Preparing data
  - Cleaning and stemming
  - Tokenizing with NLTK
- Examples and tools
  - Sentiment analysis
  - Topic modeling
  - Word embeddings

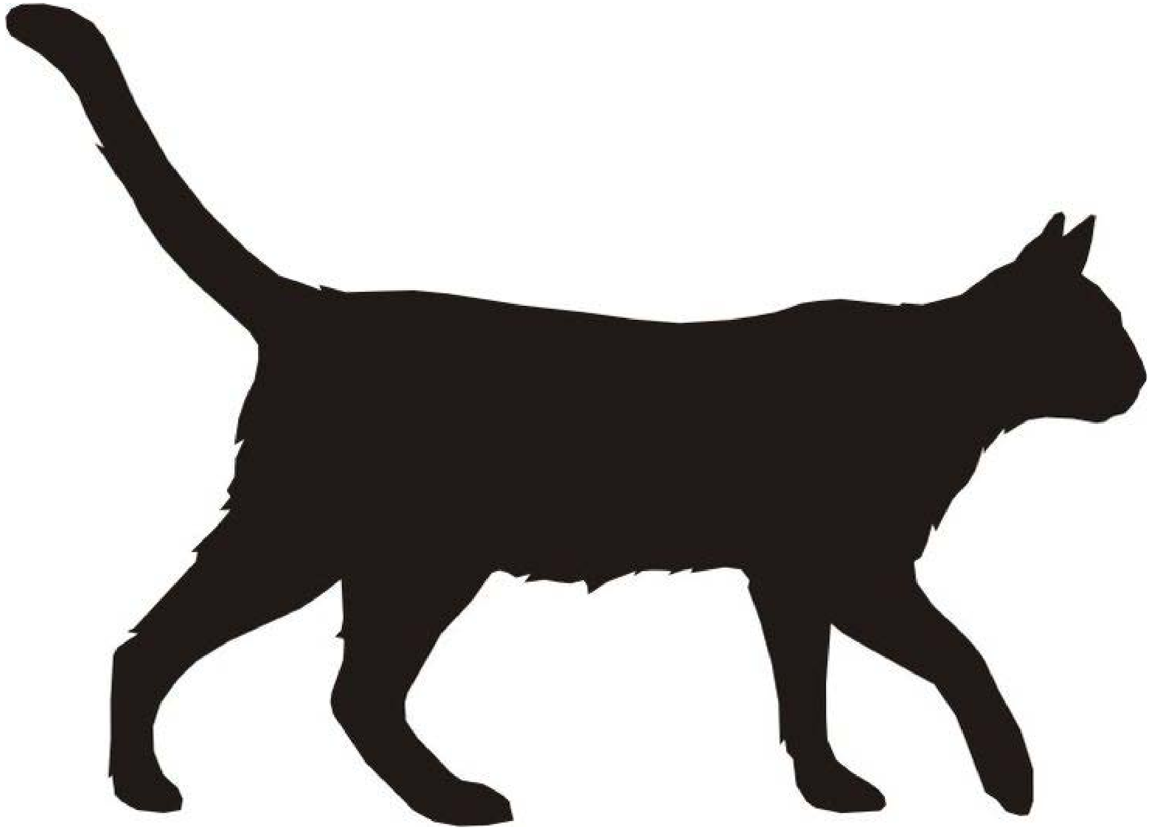
NLP is hard

- Cat, cat, cats



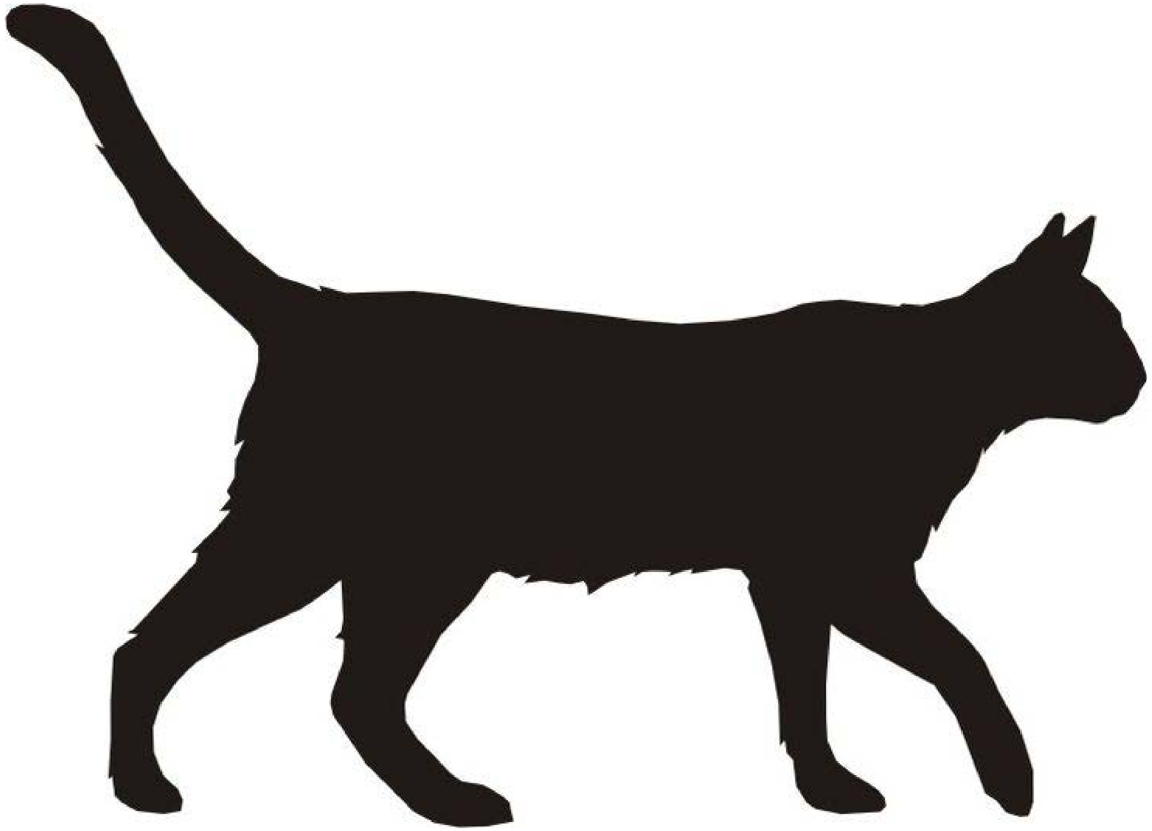
NLP is hard

- Cat, cat, cats
- catty, cattle, cataract, catacomb



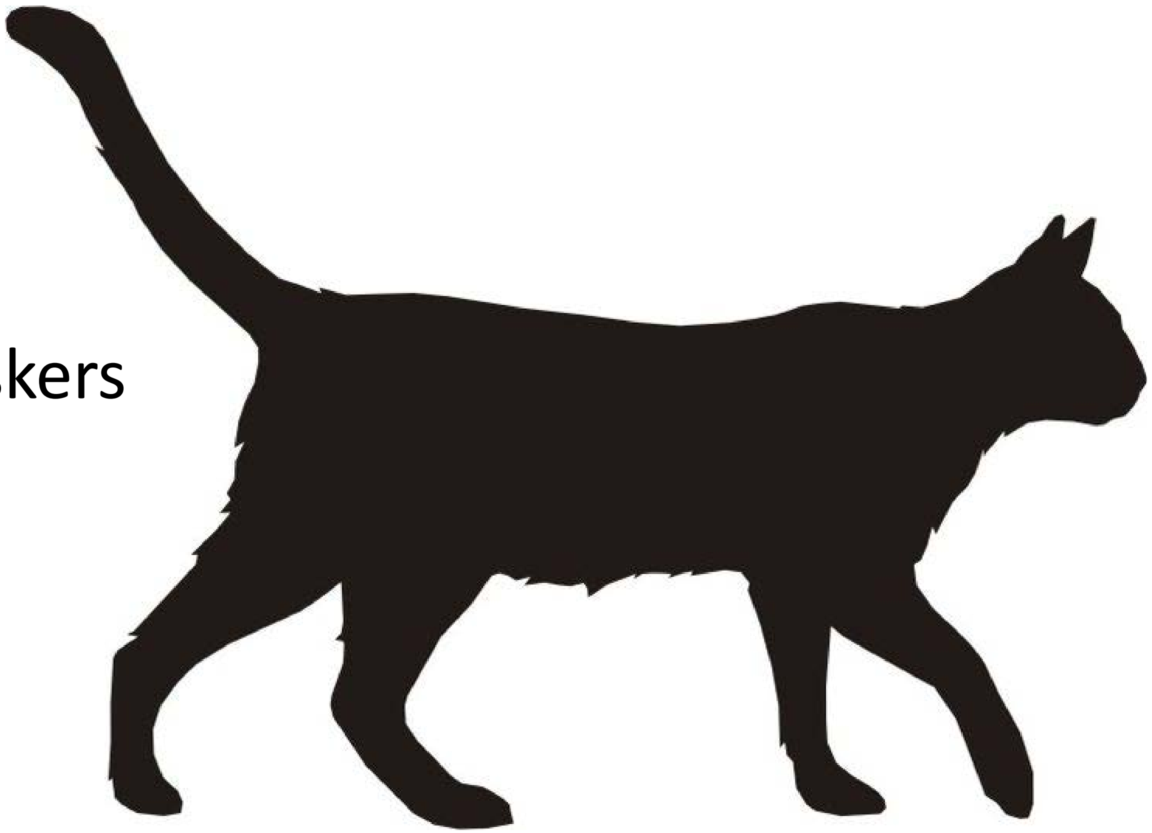
NLP is hard

- Cat, cat, cats
- catty, cattle, cataract, catacomb
- kitten, kitty, persian, tabby



NLP is hard

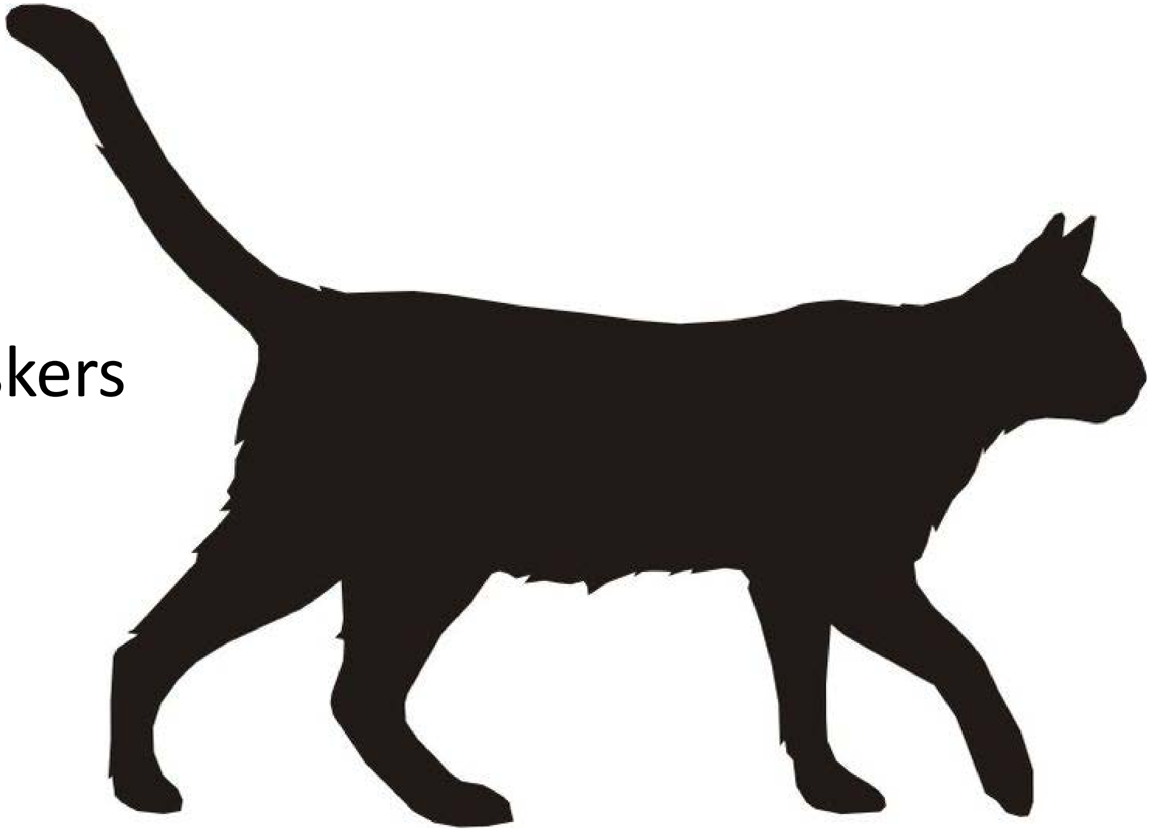
- Cat, cat, cats
- catty, cattle, cataract, catacomb
- kitten, kitty, persian, tabby
- Mittens, Tiger, Garfield, Mr. Whiskers



NLP is hard

- Cat, cat, cats
- catty, cattle, cataract, catacomb
- kitten, kitty, persian, tabby
- Mittens, Tiger, Garfield, Mr. Whiskers
- gato, chat, katze, 猫

- And that’s just *cat*



Outline

- Why is NLP hard?
- Preparing data
  - Cleaning and stemming
  - Tokenizing with NLTK
- Examples and tools
  - Sentiment analysis
  - Topic modeling
  - Word embeddings

Preparing Data – Cleaning

- As we can see, real data are very messy.
- There are a few common strategies that can help a lot
- Simple cleaning:
  - Removing punctuation
  - Lowercasing
- Stemming:
  - run/runs/running -> run

Preparing Data - Tokenization

- Tokenization is an extremely important aspect of real NLP
- It’s often critical to break a document down into sentences
  - See spot run. Run spot run. -> [‘See spot run’, ‘Run spot run’]

Preparing Data - Tokenization

- Tokenization is an extremely important aspect of real NLP
- It’s often critical to break a document down into sentences
  - See spot run. Run spot run. -> [‘See spot run’, ‘Run spot run’]
  - Dr. Radev got his Ph.D. from Columbia University in N.Y.C.

Preparing Data - Tokenization

- Tokenization is an extremely important aspect of real NLP
- It’s often critical to break a document down into sentences
  - See spot run. Run spot run. -> [‘See spot run’, ‘Run spot run’]
  - Dr. Radev got his Ph.D. from Columbia University in N.Y.C.
- It’s almost always critical to break a document down into words
  - How do you handle contractions like “don’t”?
  - How do you handle “Ph.D.”? “N.Y.C.”?
- This is where the natural language toolkit (NLTK) comes in

Preparing Data - NLTK

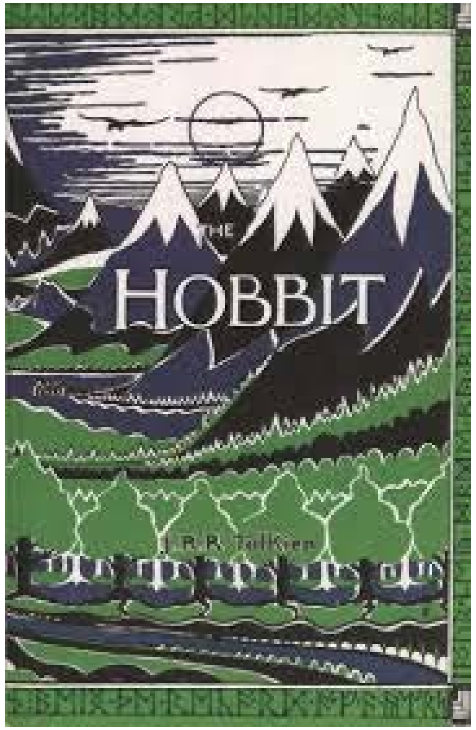
- NLTK has a wide variety of NLP tools, including a straightforward connection to tools from many other NLP groups such as Stanford
- I won't get into details, but using most of these tools can be reduced to just a few lines of Python with NLTK.

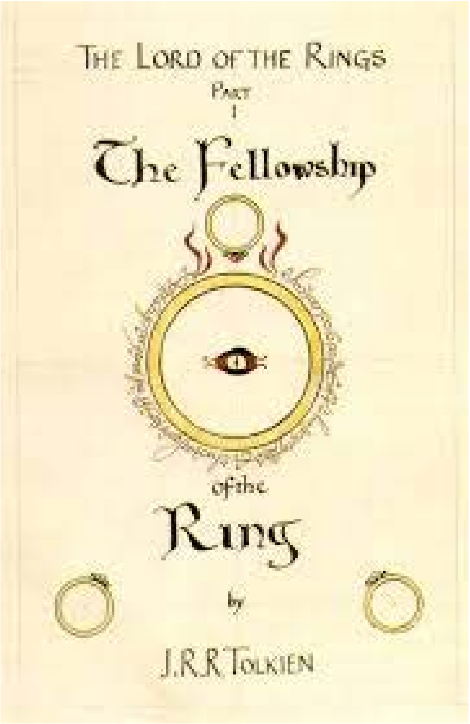
- I highly recommend NLTK

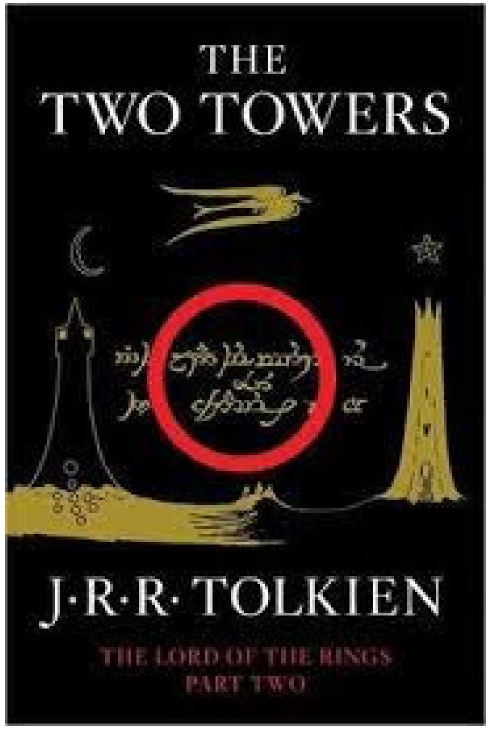
Outline

- Why is NLP hard?
- Preparing data
  - Cleaning and stemming
  - Tokenizing with NLTK
- Examples and tools
  - Summarizing a dataset
  - Sentiment analysis
  - Topic modeling
  - Word embeddings

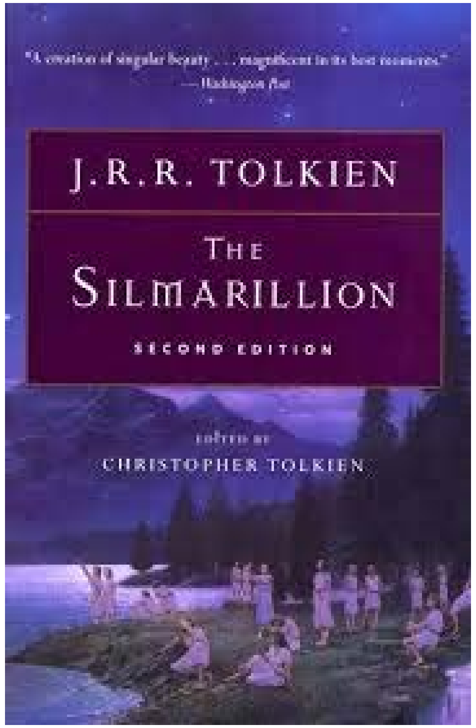
The Data Set


THE HOBBIT
J.R.R. Tolkien


The Lord of the Rings
Part I
The Fellowship of the Ring
by
J.R.R. Tolkien


THE TWO TOWERS
J·R·R· TOLKIEN
THE LORD OF THE RINGS
PART TWO

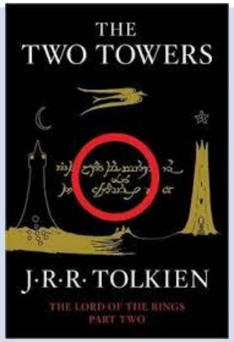
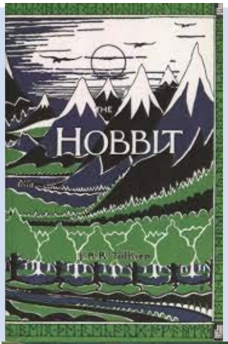
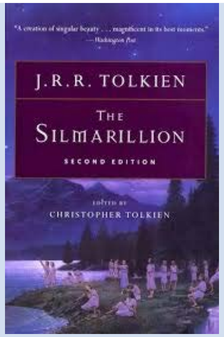
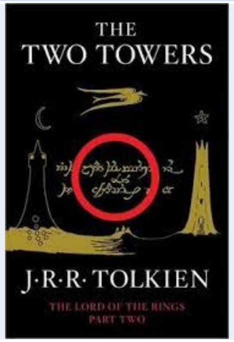
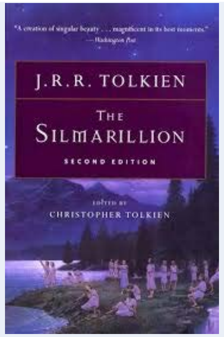
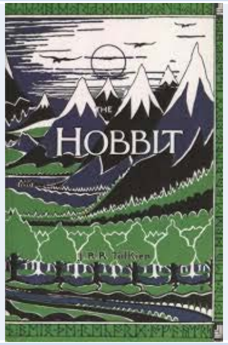
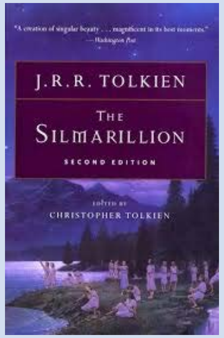
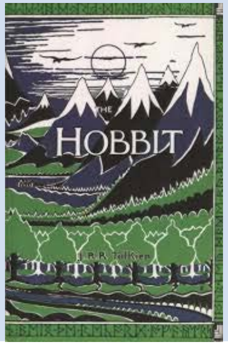
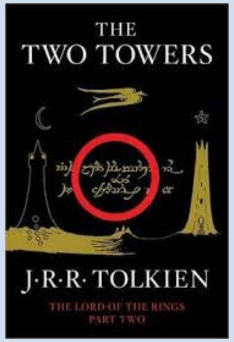

THE RETURN
OF THE KING


J.R.R. TOLKIEN
THE SILMARILLION
SECOND EDITION
EDITED BY
CHRISTOPHER TOLKIEN

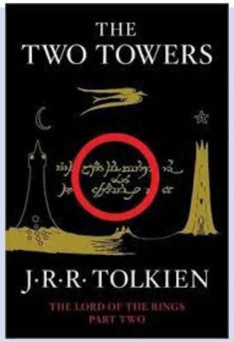
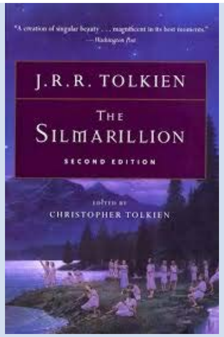
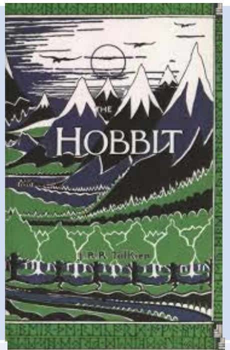
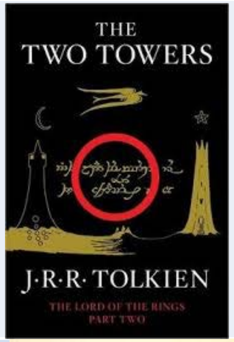
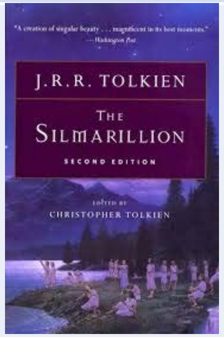
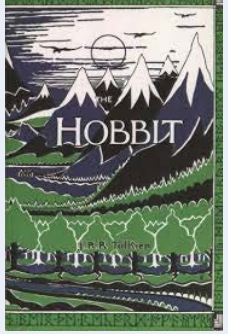
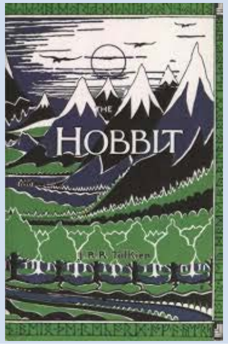
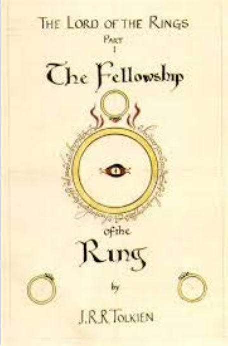
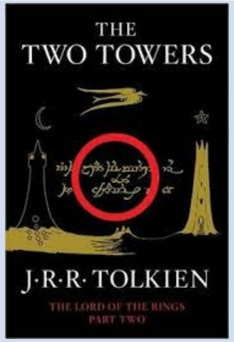
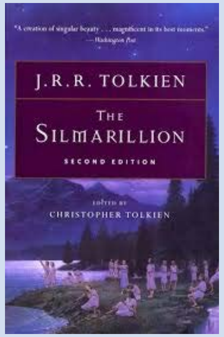
Summary Statistics

- NLP is **heavily** data-driven
- Think about how long it takes children to learn language
- Depending on the sophistication, you may require hundreds or thousands of documents to be able to use modern NLP tools
- As humans, we will need some kind of summary statistics to understand a corpus of this magnitude

Summary Statistics - Example

	Most				**Fewest**
Number of Sentences					
Number of Words (tokens)					
Tokens per Sentence					

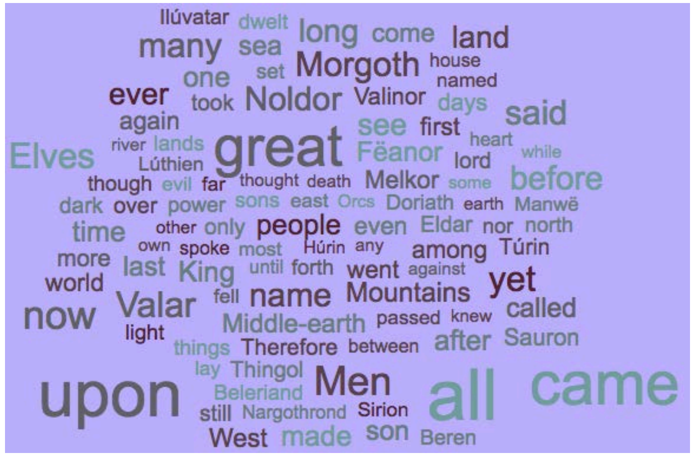
Summary Statistics - Example

	Most				Fewest
Number of Tokens					
Number of Unique Words (types)					
Types per Token					

Summary Statistics - Takeaway

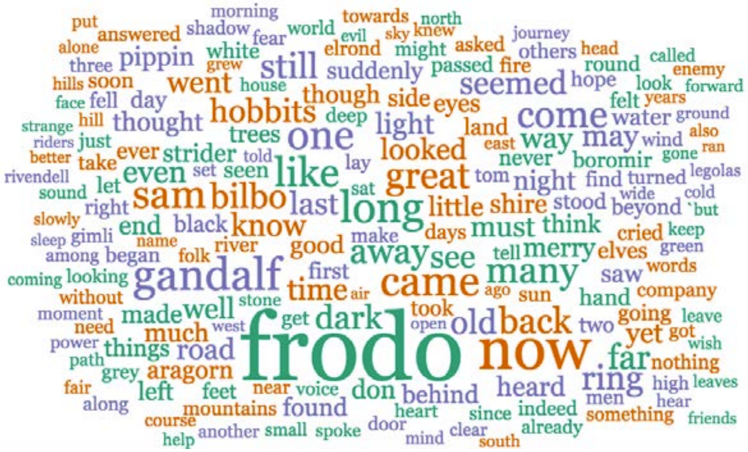
- Words/sentence can give a reasonable measure of language complexity
- Types/token can give a decent measure of vocabulary breadth
- These results depend heavily on cleaning and tokenization!

Word It Out

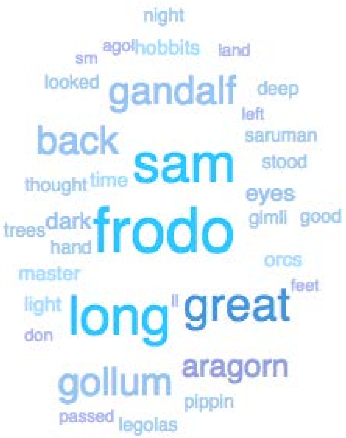


Jason Davies

Word Sift



Google Docs Add-On



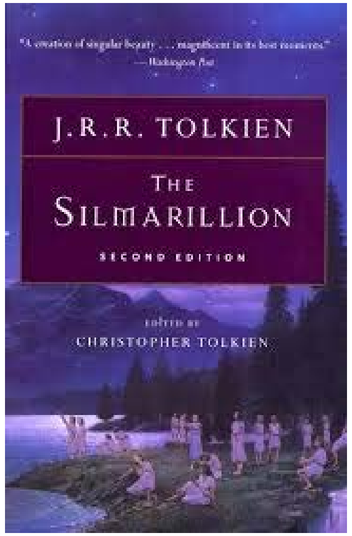
Daniel Soper

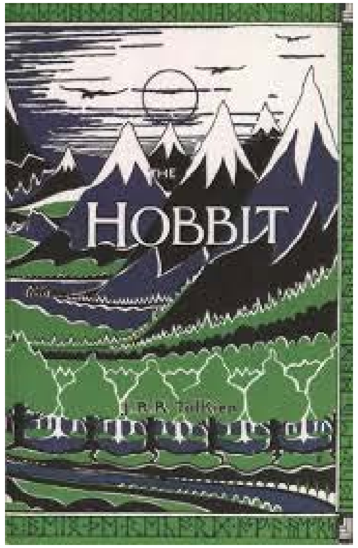
Named Entity Recognition

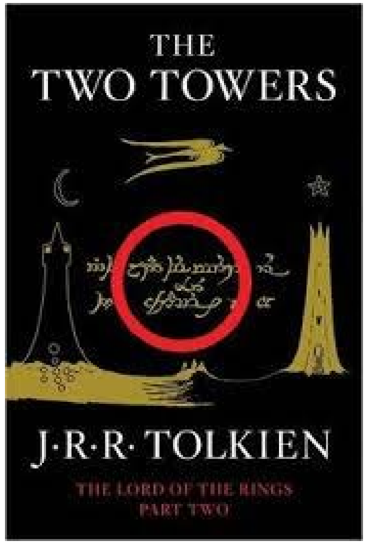
- NER tools allow you to extract entities present in a text

- PERSON, ORGANIZATION, LOCATION (MUC3)
- TIME, DATE, MONETARY VALUE, PERCENTAGE (MUC7)

Named Entity Recognition - Example







Sauron - 202	Bilbo - 527	Frodo - 995	Frodo - 464	Sam - 426
Morgoth - 187	Thorin - 229	Sam - 375	Sam - 408	Frodo - 346
Beren - 163	Balin - 67	Bilbo - 278	Gimli - 184	Pippin - 220
Eldar - 142	Baggins - 59	Strider - 192	Legolas - 163	Faramir - 149
Túrin - 112	Bard - 50	Pippin - 164	Pippin - 154	Rohan - 86

Named Entity Recognition - Takeaway

- I suggest the Stanford tool and NTLK
- Important to batch process
  - Run time went from 10 days to 5 minutes
- After you identify all the entities, you may need to combine some
  - Bilbo, Baggins, Bilbo Baggins
  - Strider, Aragorn
- As always, there will be errors
  - Shadowfax saw Gandalf (tagged as one entity)

Sentiment Analysis

- Sentiment analysis is one of the major applications of current NLP technology.

10/21/2015

1 check-in

Zingermans was recommended by a friend of mine who went to the University of Michigan for her undergrad, and boy am I glad that I listened to her!

3186 out of 4960 people found the following review useful:

It is not a sequel, but a remake

Author: sonofhades (sonofhades@hotmail.com) from Earth, 3rd planet of system Sol

16 December 2015

***** This review may contain spoilers *****

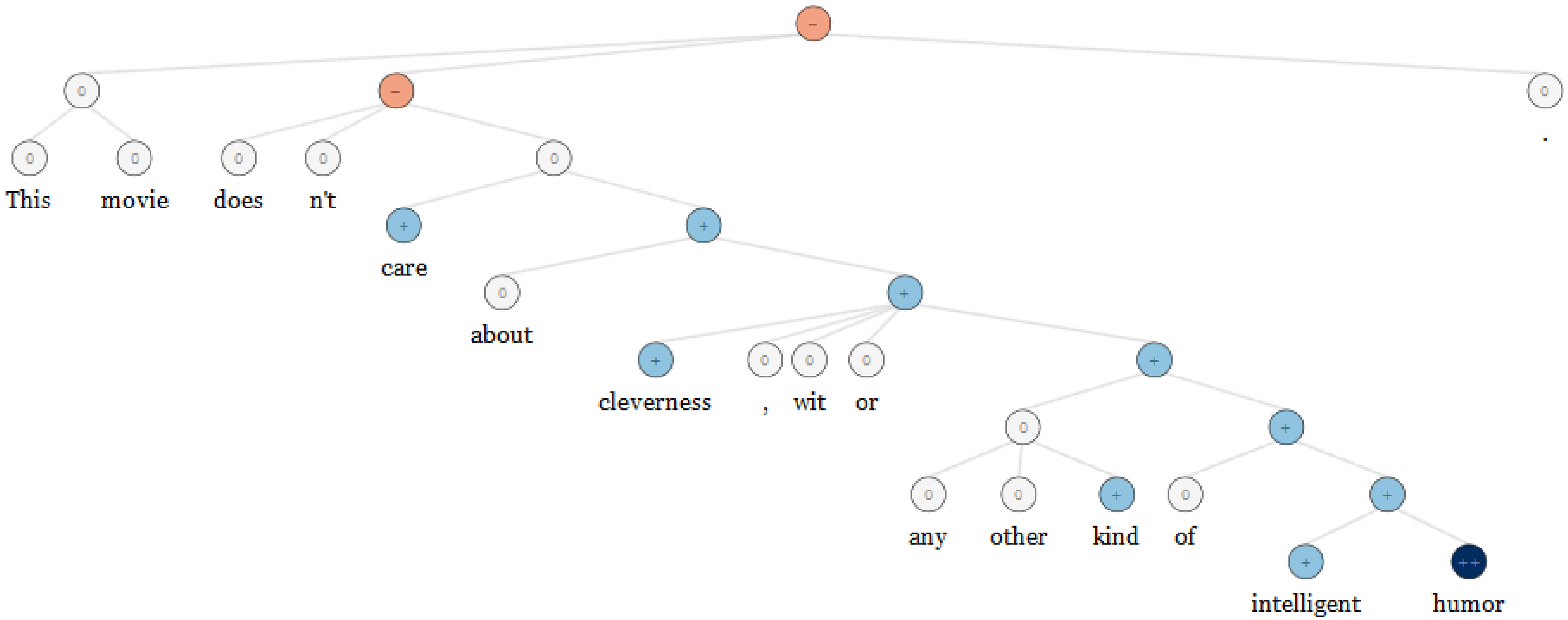
Howl at the Heavens!

5.0 out of 5 stars on April 24, 2013

This shirt has changed my life! Before, I couldn't walk through the aisles at Wal-Mart, graze on the buffet at Sizzler, or even take in a round at my local miniature golf course, without people pointing and saying, "Hey, you're that Zulu guy from Star Wars, aren't you?" Even if I wore sunglasses, I'd still get mistaken for Yoko Ono.

Sentiment Analysis

- Sentiment analysis is one of the major applications of current NLP technology.
- The field has recently seen strong advances due to Deep Learning.



Sentiment Analysis - Example

	Highest							**Lowest**
Overall Average Sentiment								
Sentiment Standard Deviation								

Sentiment Analysis - Takeaway

- I suggest the new Stanford tool
- Be wary of domain differences!
  - She's a great athlete and she was not afraid to be aggressive.
  - This is a terrible restaurant. The wait staff were very aggressive.
  - Best to have a model that is trained on the same domain

Topic Modelling

- Topics models are a great way to explore a corpus
- Generative model of document creation
  - Each document is a weighted combination of topics
  - Each topic is a weighted combination of words
  - All words appear in all topics with some (small) probability
- To add a word to a document
    - Pick a topic according to the documents weighted composition
    - Pick a word according to that topic’s weighted composition
    - Add the chosen word
- LDA is one of several methods for reversing this process to discover the topics that make up a document

Topic Modelling - Example

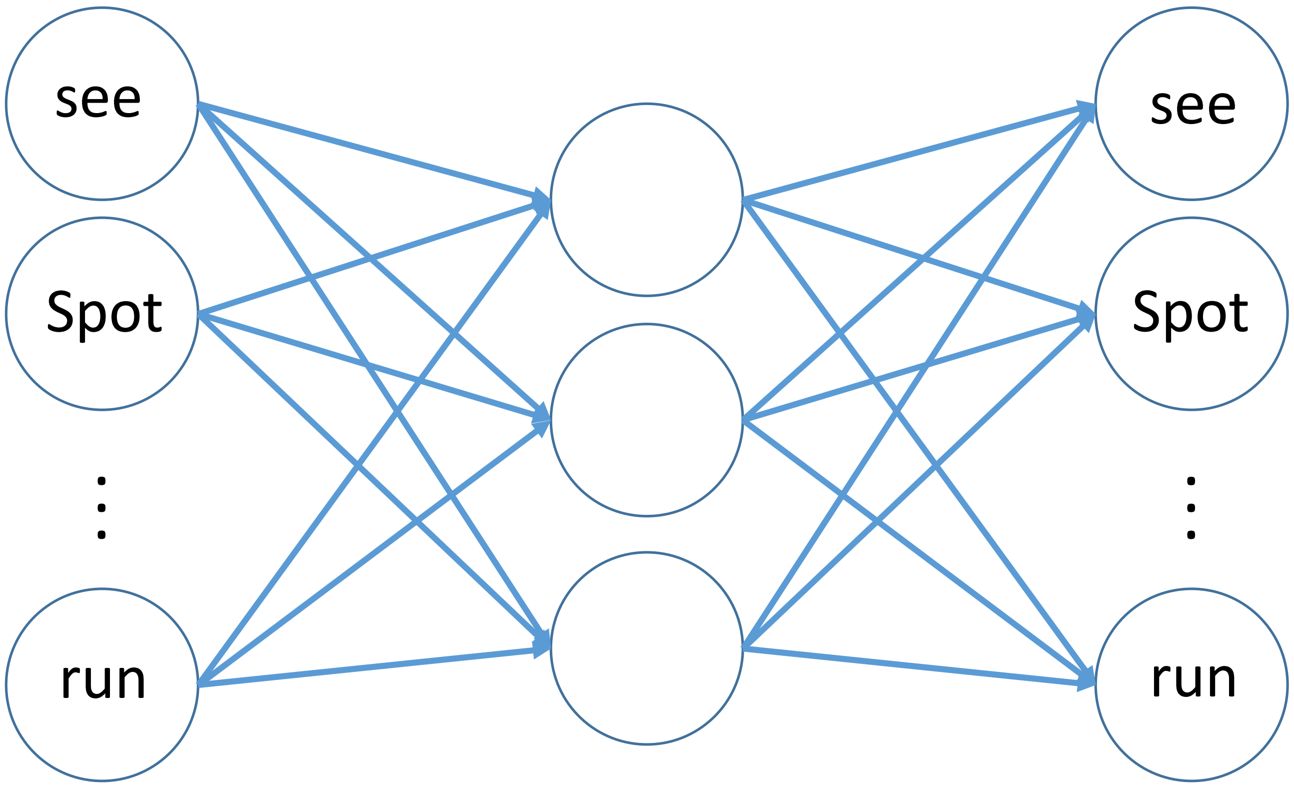
1. 0.009*upon + 0.008*away + 0.007*came + 0.007*lay
2. 0.007*now0.034*said + 0.017*n't + 0.012*Sam + 0.012*will + 0.011*Frodo
3. 0.011*came + 0.011*'I + 0.008*long + 0.008*great + 0.007*Orcs
4. 0.010*eyes + 0.008*great + 0.008*looked + 0.008*Sam + 0.008*seemed
5. 0.010*great + 0.006*name + 0.005*Morgoth + 0.005*strength + 0.005*power

Topic Modelling - Takeaway

- Straightforward, though somewhat tedious, with Gensim
- In my opinion, not reliable for classification but good for exploration
- Not all topics will be logical for a human
- Results strongly depend on number of topics (hyperparameter)

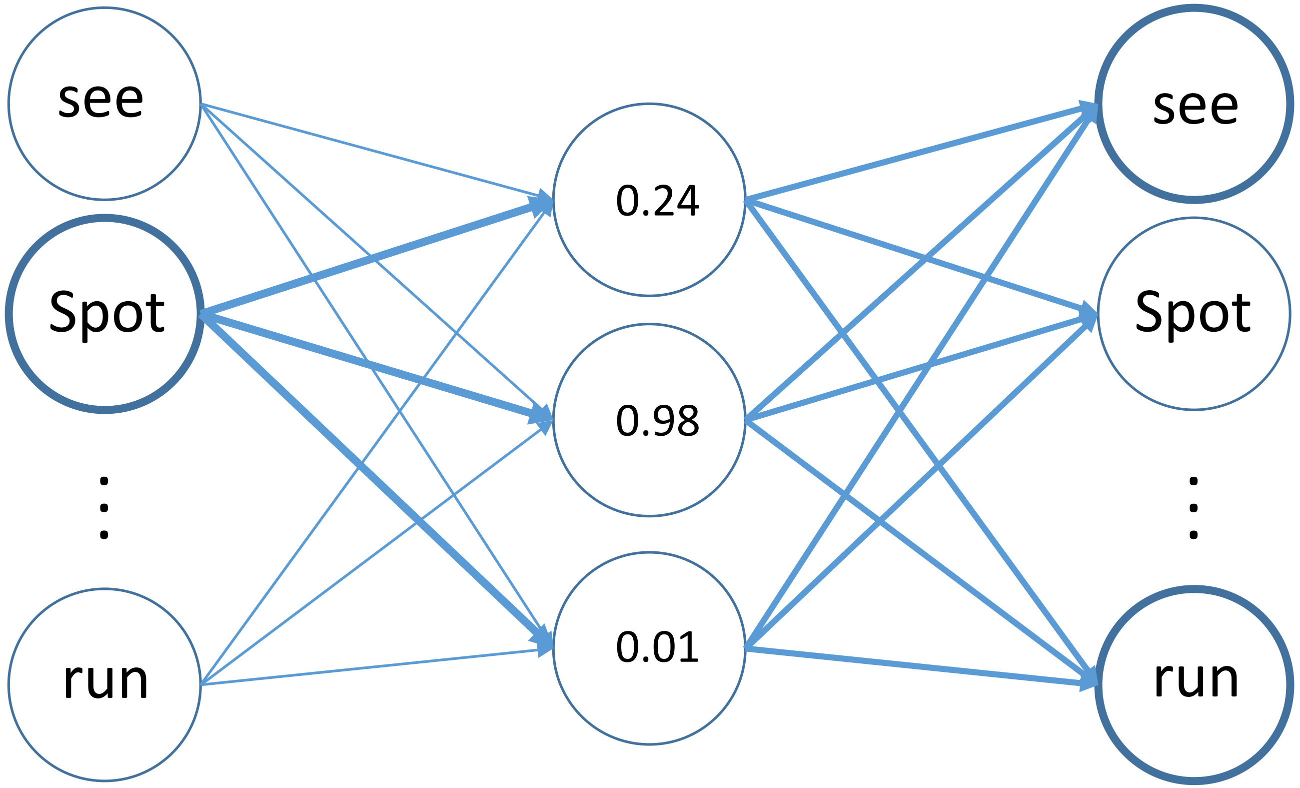
Word Embeddings

- Gensim's Word2Vec is a great tool for generating word embeddings



Word Embeddings

- Gensim's Word2Vec is a great tool for generating word embeddings



Word Embeddings – Example uses

- One of these things doesn’t belong
  - [Bilbo, Frodo, Sam, Merry, Pippin] -> Bilbo
- Numerical similarity of word pair
  - (ghost, spirit) -> 0.711402184978
- Most similar words
  - bread -> butter, cream, hot, dried
  - lembas -> mastery, maker, waybread, Dragons

Word Embeddings - Takeaway

- Flexible, useful way to represent word semantics
- Lots of pretrained models available for download
- Best to train your own, provided you have enough data
    - You may need quite a bit of data

NLP and You

- Modern tools make it very practical to include NLP in any project
- NLTK and Gensim are good tools focused on simplicity and easy of use
- All the code I wrote for my analysis is available on GitHub, complete with a wiki to help you install support tools
  - Github name: reedcoke
- Feel free to contact me with any questions – reedcoke@umich.edu