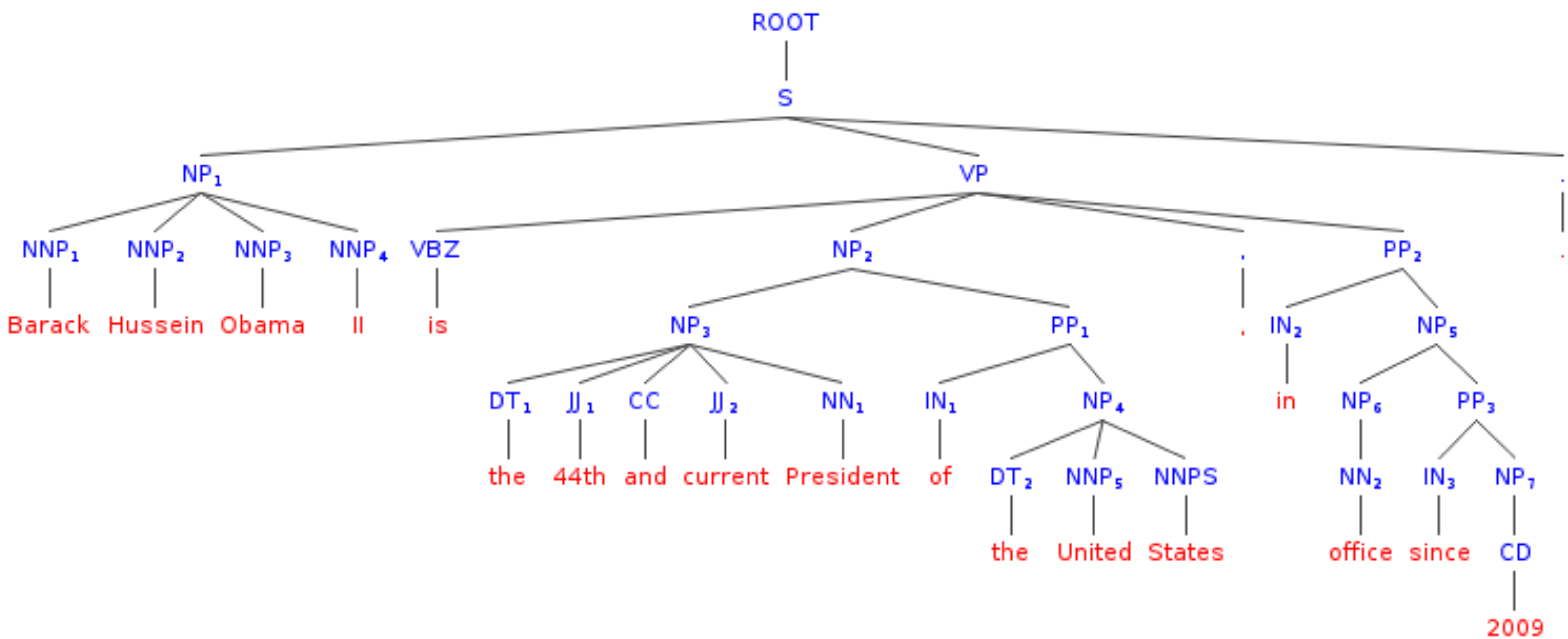
An Introduction to Natural Language Processing

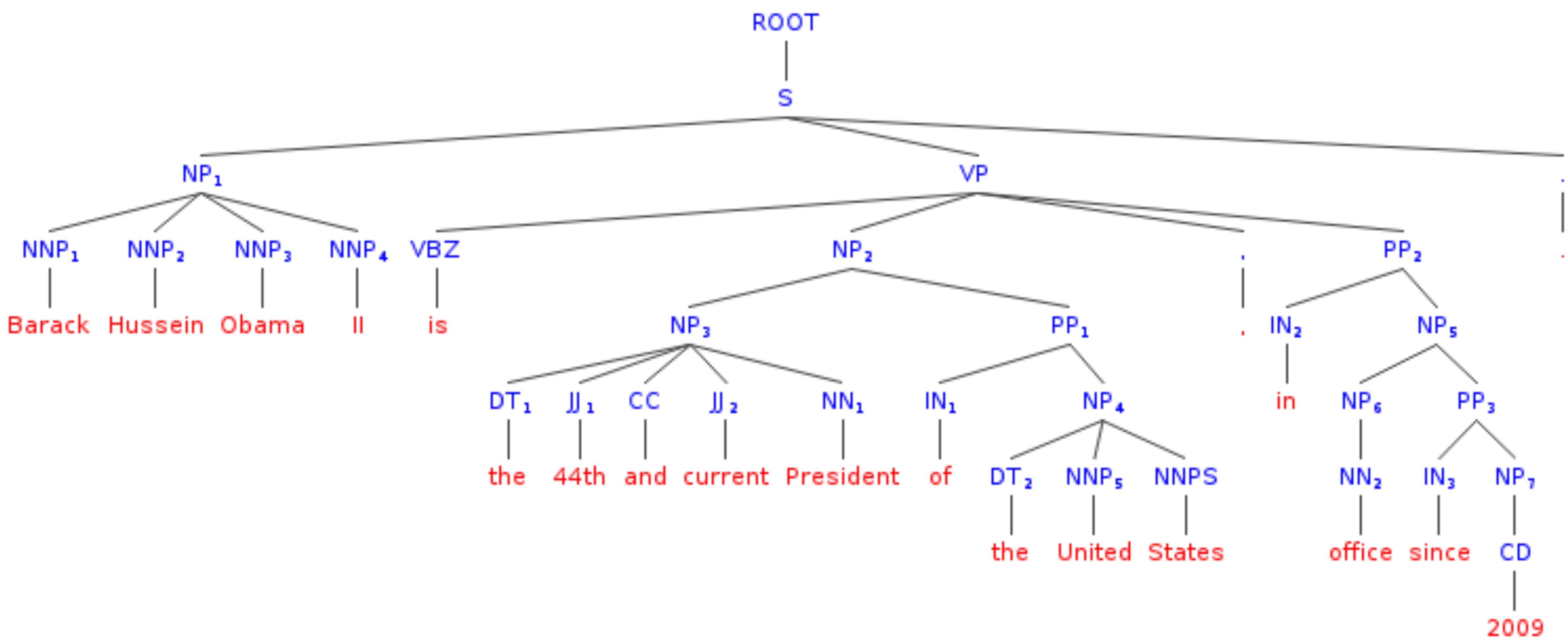
Diane M. Napolitano
Educational Testing Service

What is Natural Language Processing?

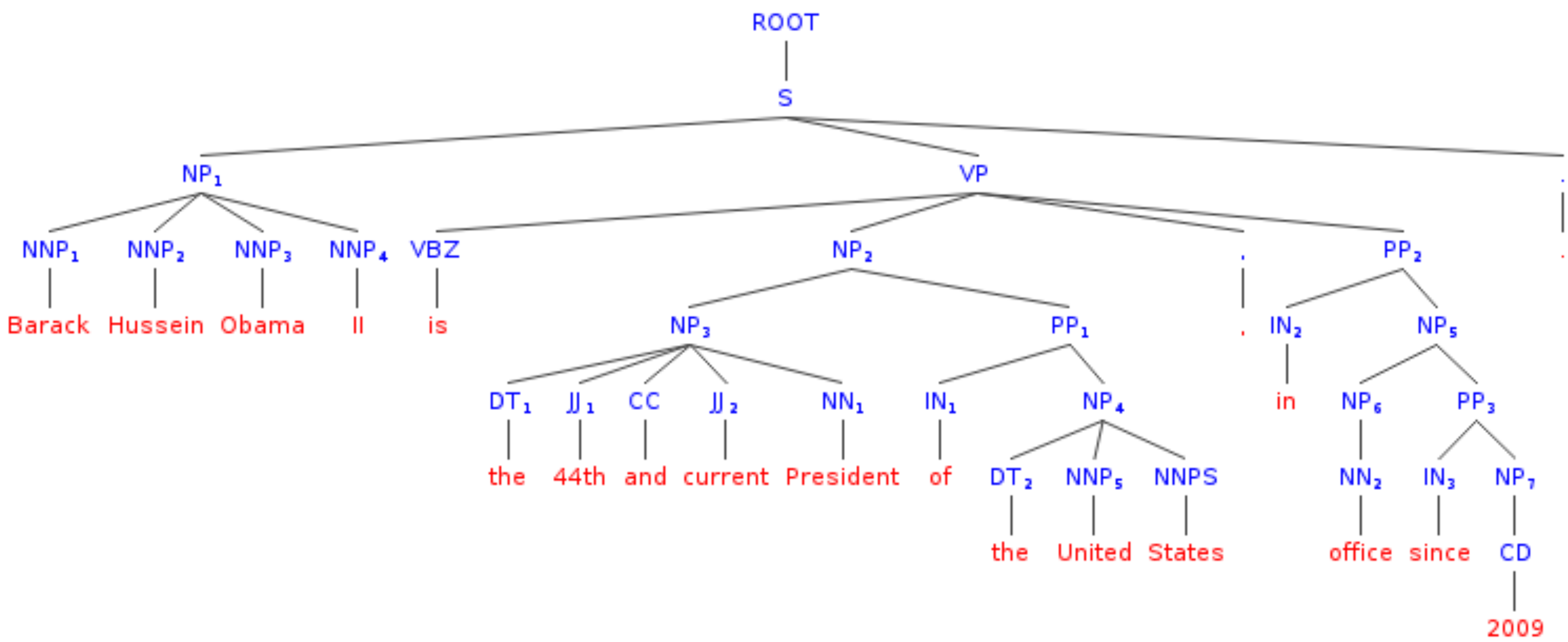
- Linguistics + Computer Science + Artificial Intelligence
- Different from, but related to, Computational Linguistics
  - Computational Linguistics is **science**
    - Study of language from a "computational perspective"
  - Natural Language Processing is **engineering**
    - Uses computers to do "useful things with language"


ROOT
S
NP1
VP
.
NNP1
NNP2
NNP3
NNP4
VBZ
NP2
,
PP2
.
Barack
Hussein
Obama
II
is
NP3
PP1
,
IN2
NP5
DT1
JJ1
CC
JJ2
NN1
IN1
NP4
in
NP6
PP3
the
44th
and
current
President
of
DT2
NNP5
NNPS
NN2
IN3
NP7
the
United
States
office
since
CD
2009

Introduction



Stanford Parser + phpSyntaxTree



Stanford Parser + phpSyntaxTree

[ROOT [S [NP [NNP Barack] [NNP Hussein] [NNP Obama] [NNP II]] [VP [VBZ is] [NP [NP [DT the] [JJ 44th] [CC and] [JJ current] [NN President]] [PP [IN of] [NP [DT the] [NNP United] [NNPS States]]]] [, ,] [PP [IN in] [NP [NP [NN office]] [PP [IN since] [NP [CD 2009]]]]]] [. .]]]

NLP vs. Information Retrieval vs. Information Extraction

- **Information Retrieval (IR)** aims to provide us with relevant documents matching a certain query
- **Information Extraction (IE)** aims to extract certain pieces of information from individual documents
- Both use NLP to complete many tasks, including…

NLP vs. Information Retrieval vs. Information Extraction

- **Information Retrieval (IR)** aims to provide us with relevant documents matching a certain query

Google

- **Information Extraction (IE)** aims to extract certain pieces of information from individual documents

- Both use NLP to complete many tasks, including...

Named Entity Recognition

[ROOT [S [NP [NNP Barack] [NNP Hussein] [NNP Obama] [NNP II]] [VP [VBZ is] [NP [NP [DT the] [JJ 44th] [CC and] [JJ current] [NN President]] [PP [IN of] [NP [DT the] [NNP United] [NNPS States]]]] [, ,] [PP [IN in] [NP [NP [NN office]] [PP [IN since] [NP [CD 2009]]]]]] [. .]]]

Barack Hussein Obama II	PERSON
United States	LOCATION
2009	DATE

Some Other NLP Tasks

- Word Sense Disambiguation

"Which airlines **serve** Denver?"
vs.
"Which airlines **serve** breakfast?"

- Coreference Resolution: Who is "he" and what is "it"?

"John went to Bill's car dealership to check out an Acura Integra. He looked at it for about an hour."

Some Other NLP Tasks

- Sentiment Analysis: How does the author feel about the topic?
  - Difficult in domains where sarcasm is often used (e.g. movie reviews) and when phrases aren't always used sarcastically

A: My basketball team just won the high school championships!
B: Yeah, right.
A: Yeah, right! I'm so glad you understand!

Source: Urban Dictionary

Some Applications of NLP

- Automated Essay/Spoken Response/Short Response Scoring
- OCR
- Automatic Summarization

We found **9** documents about atrial fibrillation.

Here are some links to more specific subtopics of atrial fibrillation: definition — symptoms — causes — diagnosis — treatment — complications — prevention — prognosis

Here's some general information on atrial fibrillation put together from the 9 documents: Atrial fibrillation is a common type of palpitation, where you experience an irregular and often rapid beating of the upper chambers of the heart, known as the atria. There are a number of treatment options for AF. The first line of treatment usually involves medications, but there are other treatments which may be appropriate. Symptoms of AF may include one or more of the following: heart palpitations, lack of energy or feeling over-tired, etc. The causes of atrial fibrillation include: rheumatic heart disease, ischaemic heart disease, etc. Treatment varies from case to case and your general outlook will depend on the severity of your underlying heart condition. Medications are prescribed in the management of atrial fibrillation depending on the overall treatment goal: Heart surgery Patients at "low risk" may be given aspirin 325 mg/d to prevent stroke. See also: Anticoagulants, Arrhythmias, etc.

Source: Nenkova, Ani and McKeown, Kathleen. *Automatic Summarization.* 2011.

Some Applications of NLP

- Automated Essay/Spoken Response/Short Response Scoring
- OCR
- Automatic Summarization

We found **9** documents about atrial fibrillation.

Here are some links to more specific subtopics of atrial fibrillation: definition — symptoms — causes — diagnosis — treatment — complications — prevention — prognosis

Here's some general information on atrial fibrillation put together from the 9 documents: Atrial fibrillation is a common type of palpitation, where you experience an irregular and often rapid beating of the upper chambers of the heart, known as the atria. There are a number of treatment options for AF. The first line of treatment usually involves medications, but there are other treatments which may be appropriate. Symptoms of AF may include one or more of the following: heart palpitations, lack of energy or feeling over-tired, etc. The causes of atrial fibrillation include: rheumatic heart disease, ischaemic heart disease, etc. Treatment varies from case to case and your general outlook will depend on the severity of your underlying heart condition. Medications are prescribed in the management of atrial fibrillation depending on the overall treatment goal: Heart surgery Patients at "low risk" may be given aspirin 325 mg/d to prevent stroke. See also: Anticoagulants, Arrhythmias, etc.

Query for "atrial fibrillation", receive summary of nine relevant documents, generated on-the-fly.

Source: Nenkova, Ani and McKeown, Kathleen. *Automatic Summarization.* 2011.

Syntactic Linguistic Units

"That that is, is. That that is not, is not. Is that it? It is."

Words	"That", "that", "is", "is", "That", "that", "is", "not", "is", "not", "Is", "that", "it", "It", "is"	15
Tokens	"That", "that", "is", "is", ".", "That", "that", "is", "not", ",", "is", "not", ".", "Is", "that", "it", "?", "It", "is", "."	20
Types	"that", "is", ",", ".", "not", "it", "?"	7

Stems and Lemmas

- "run", "running", and "ran" are all different types, but ultimately they are different *inflicted* forms of the same word
- **stemming** uses rules to remove word affixes
- **lemmatization** uses a resource such as WordNet to find the root word of an inflicted form

	run	running	ran
Stem	run	run	ran
Lemma	run	run	run

Tokenization

- Pretty straightforward if we know the **sentence boundaries**
  1. It was due Friday by 5 p.m. Saturday would be too late.
  2. She has an appointment at 5 p.m. Saturday to get her car fixed.

- And also if our text is clean

Tokenization

```
Date: Sat, 21 Jun 2003 03:44:38 -0700 (PDT)
From: Merrick Berg <mmmmberg@yahoo.com>
To: lars@winds.gsfc.nasa.gov
Cc: Nilani <anilani@cs.umd.edu>
Subject: computer

Hi Lars:
Call me around 4-5 p.m.
if you would like to come over
this evening and configure your computer.I'll be
looking forward to see you tomorrow.
There are things that I don't understand in Treemap
that you may already know having spent so much time on
it.
Maybe you could tell me
those things right away rather than me
trying to understand spending too much time
on them.
```

Word-Level Tokenization

- A process known as "exploding punctuation" at ETS 😄

- Adhere to the **Penn Treebank**

"I", "thought", ",", "``", "Yeah", "right", ",", "come", "tell", "me", "about", "it", "!", "''"

- Tokens defined by **semantics**

1. "She", "has", "an", "appointment", "at", "5", "p.m.", "Saturday", "to", "get", "her", "car", "fixed", "."
2. "I", "would", "n't", "go", "in", "there", "if", "I", "were", "you", "."

Additional (Syntactic) Units: The N-Gram

- unigrams, bigrams, trigrams, 4-grams, 5-grams...
- Can provide additional contextual information necessary for complex natural language problems
  - "You shall know a word by the company it keeps." - J.R. Firth

bigrams	trigrams	4-grams
"you shall"	"you shall know"	"you shall know a"
"shall know"	"shall know a"	"shall know a word"
"know a"	"know a word"	"know a word by"
...	...	...

Lots of N-Grams ➡️ A Model of Language

- **Language Model**: A collection of n-grams from *at least one very large corpus* and the frequency of their occurrence within it
  - Can provide information about how *common* or *rare* words or phrases are
  - **Conditional Frequency**: the number of occurrences of that n-gram type over the total number of n-gram tokens

A Bigram Language Model of the Brown Corpus

```
from __future__ import division

import re

from nltk.corpus import brown
from nltk.corpus import stopwords
from nltk.util import ngrams


stopwords_list = stopwords.words('english')

bigrams = ngrams([w.lower() for w in brown.words()], 2)
content_bigrams = [b for b in bigrams if
                   (re.search(r'^\w+$', b[0]) and re.search(r'^\w+$', b[1]))
                   and
                   (b[0] not in stopwords_list and b[1] not in stopwords_list)]

for ngram in set(content_bigrams):
    conditional_freq = content_bigrams.count(ngram) / len(content_bigrams)
    print "\t".join([str(ngram), str(conditional_freq)])
```

A Bigram Language Model of the Brown Corpus

```
from __future__ import division

import re

from nltk.corpus import brown
from nltk.corpus import stopwords
from nltk.util import ngrams


stopwords_list = stopwords.words('english')

bigrams = ngrams([w.lower() for w in brown.words()], 2)
content_bigrams = [b for b in bigrams if
                   (re.search(r'^\w+$', b[0]) and re.search(r'^\w+$', b[1]))
                   and
                   (b[0] not in stopwords_list and b[1] not in stopwords_list)]

for ngram in set(content_bigrams):
    conditional_freq = content_bigrams.count(ngram) / len(content_bigrams)
    print "\t".join([str(ngram), str(conditional_freq)])
```

Remove non-content words

A Bigram Language Model of the Brown Corpus

```
from __future__ import division

import re

from nltk.corpus import brown
from nltk.corpus import stopwords
from nltk.util import ngrams


stopwords_list = stopwords.words('english')

bigrams = ngrams([w.lower() for w in brown.words()], 2)
                 = [b for b in bigrams if
                   (re.search(r'^\w+$', b[0]) and re.search(r'^\w+$', b[1]))
                   and
                   (b[0] not in stopwords_list and b[1] not in stopwords_list)]

for ngram in set(content_bigrams):
    conditional_freq = content_bigrams.count(ngram) / len(content_bigrams)
    print "\t".join([str(ngram), str(conditional_freq)])
```

Remove non-content words

No bigrams of punctuation

A Bigram Language Model of the Brown Corpus

```
from __future__ import division

import re

from nltk.corpus import brown
from nltk.corpus import stopwords
from nltk.util import ngrams


stopwords_list = stopwords.words('english')

bigrams = ngrams([w.lower() for w in brown.words()], 2)
                   = [b for b in bigrams if
                      (re.search(r'^\w+$', b[0]) and re.search(r'^\w+$', b[1]))
                      and
                      (b[0] not in stopwords_list and b[1] not in stopwords_list)]

for ngram in set(content_bigrams):
    conditional_freq = content_bigrams.count(ngram) / len(content_bigrams)
    print "\t".join([str(ngram), str(conditional_freq)])
```

Remove non-content words

No bigrams of punctuation

Occurrence conditional on other bigrams in corpus

What does our language model tell us?

- 135,369 unique bigrams over 1.18 million words
- 87.5% of bigrams occur once
- Domain is too specific (American English, 1961, mostly news)
- Corpus is too small (500 samples)
    - We can compensate for sparse data
    - "It never pays to think until you've run out of data." - Eric Brill 😅

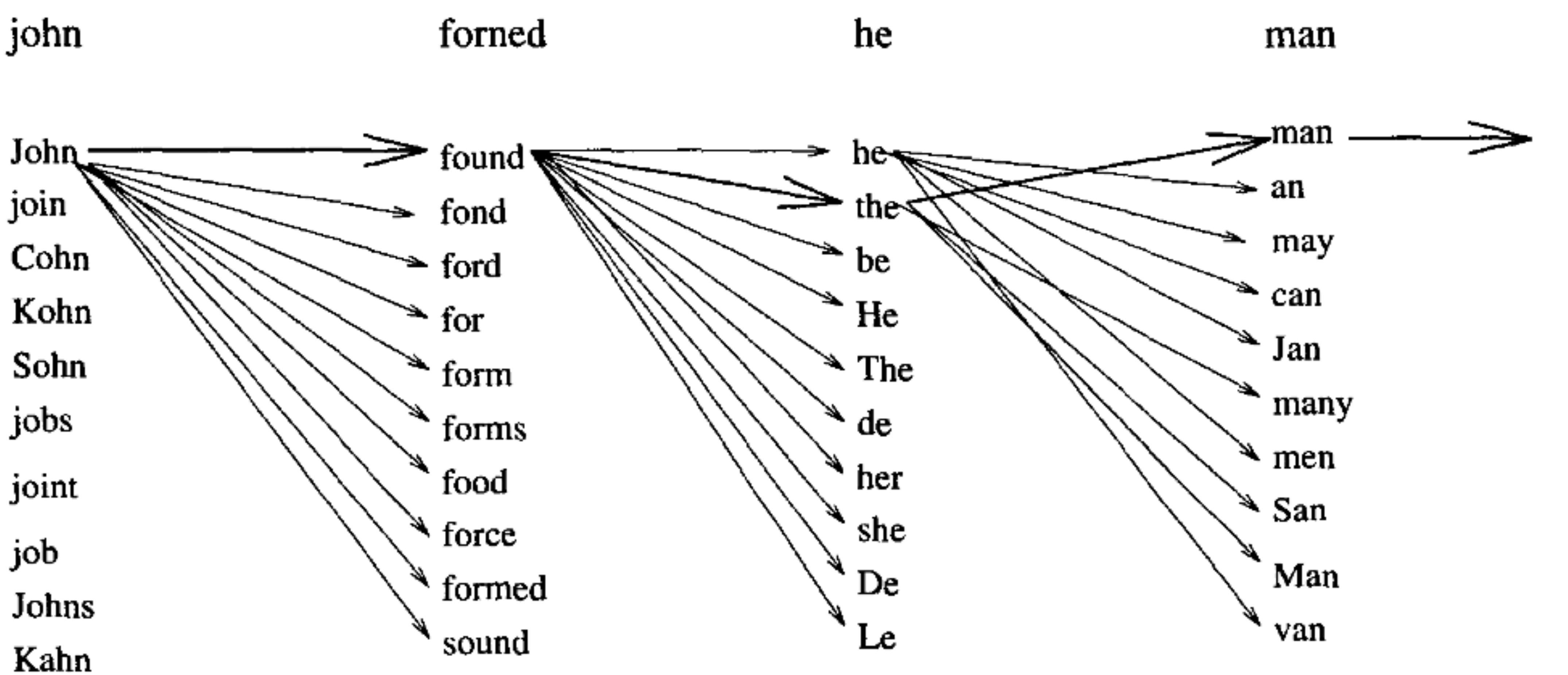
Top 10 Most Frequent:	
"united states"	0.224%
"new york"	0.170%
"per cent"	0.084%
"years ago"	0.079%
"rhode island"	0.052%
"could see"	0.050%
"last year"	0.045%
"even though"	0.044%
"high school"	0.042%
"white house"	0.040%

Using Language Models to Improve OCR

Original Sentence: John found the man.

Input Sentence: john fornd he man.

Corrected Sentence: John found the man.



Best Word Sequence: John found the man .

Source: Tong and Evans. A Statistical Approach to Automatic OCR Error Correction in Context. *ACL* 1996.

Latent Semantic Analysis/Indexing

- How *similar* are two different n-grams?
  - Latent Semantic **Analysis** when we have two n-grams and want a measure of similarity between them
  - Latent Semantic **Index** when we have one n-gram and want others that are related to it

Querying an LSI for the 20 Most-Similar Terms

- Demo: http://lsa.colorado.edu/
- Corpus: "General Reading up to 1st Year College"

"dogs"							
1	dogs	6	leash	11	snarling	16	collies
2	dog	7	huskies	12	unshopped	17	pups
3	barking	8	wagging	13	oogruk	18	puppy
4	barked	9	kennel	14	terrier	19	stray
5	collie	10	manilak	15	puppies	20	mongrel

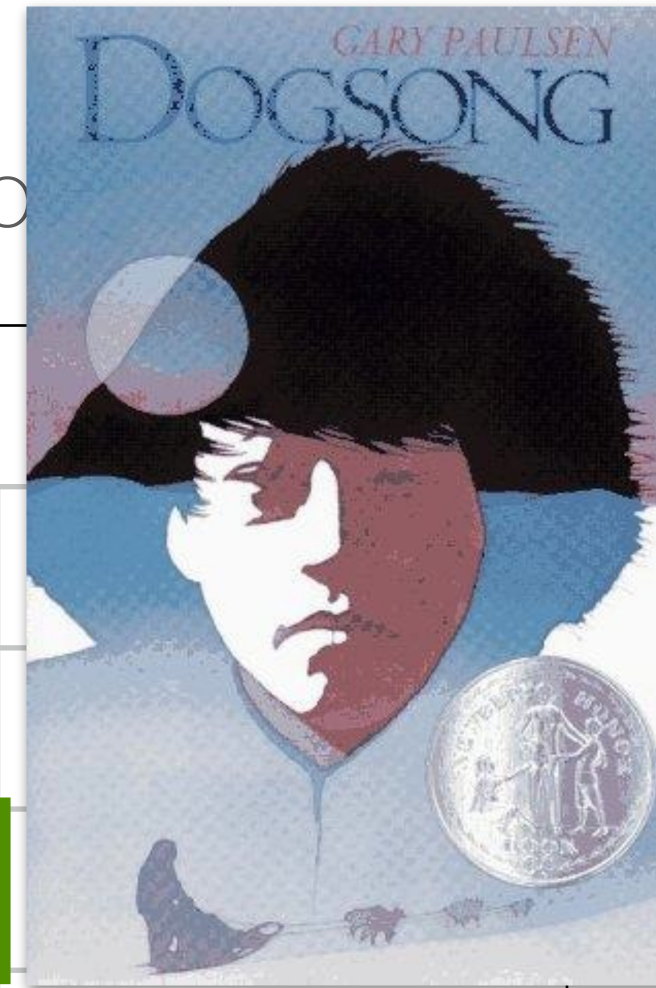
Querying an LSI for the 20 Most-Similar Terms

- Demo: http://lsa.colorado.edu/
- Corpus: "General Reading up to 1st Year College"

"dogs"							
1	dogs	6	leash	11	snarling	16	collies
2	dog	7	huskies	12	unshopped	17	pups
3	barking	8	wagging	13	oogruk	18	puppy
4	barked	9	kennel	14	terrier	19	stray
5	collie	10	manilak	15	puppies	20	mongrel

Querying an LSI for the 20 Most-Similar Terms

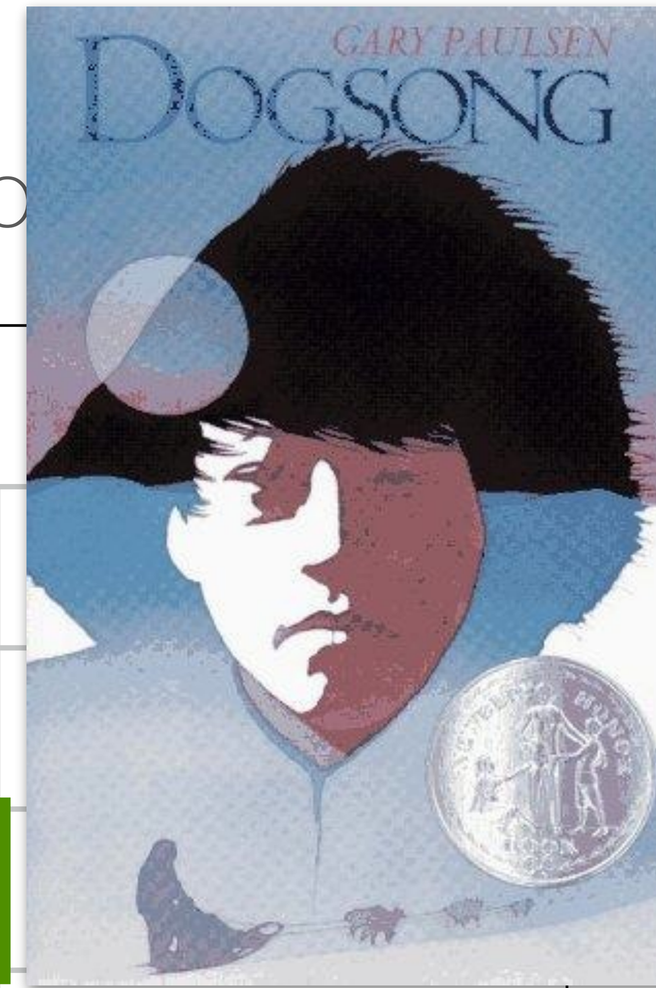
- Demo: http://lsa.colorado.edu/
- Corpus: "General Reading up to 1st Year Co



"dogs"							
1	dogs	6	leash	11	snarling		
2	dog	7	huskies	12	unshopped		
3	barking	8	wagging	13	oogruk		
4	barked	9	kennel	14	terrier	19	stray
5	collie	10	manilak	15	puppies	20	mongrel

Querying an LSI for the 20 Most-Similar Terms

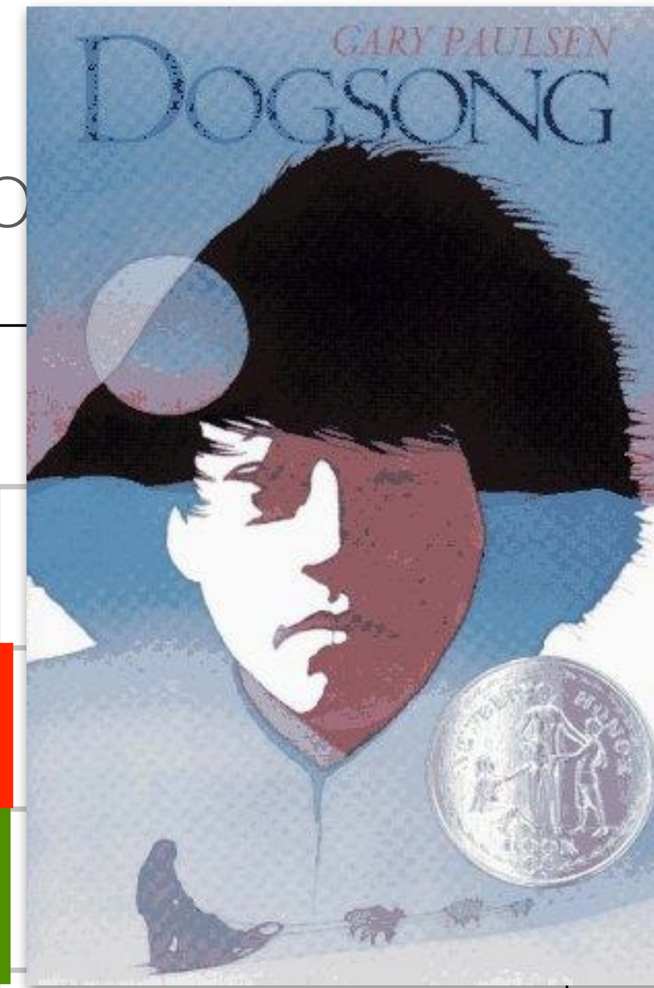
- Demo: http://lsa.colorado.edu/
- Corpus: "General Reading up to 1st Year Co



"dogs"							
1	dogs	6	leash	11	snarling		
2	dog	7	huskies	12	unshopped		
3	barking	8	wagging	13	oogruk		
4	barked	9	kennel	14	terrier	19	stray
5	collie	10	manilak	15	puppies	20	mongrel

Querying an LSI for the 20 Most-Similar Terms

- Demo: http://lsa.colorado.edu/
- Corpus: "General Reading up to 1st Year Co



"dogs"							
1	dogs	6	leash	11	snarling		
2	dog	7	huskies	12	unshopped		
3	barking	8	wagging	13	oogruk		
4	barked	9	kennel	14	terrier	19	stray
5	collie	10	manilak	15	puppies	20	mongrel

Parts of Speech

- When n-grams of tokens or words aren't enough
- Again follow the **Penn Treebank** standard

I/PRP would/MD n't/RB go/VB in/IN there/EX if/IN I/PRP were/VBD you/PRP ./.

PRP	Personal Pronoun		IN	Preposition
MD	Modal Verb		EX	Existential *there*
RB	Adverb		VBD	Verb, past tense
VB	Verb, root form		.	End of sentence

Using POS to Resolve Lemmatization Ambiguity

- By default, assume everything is a noun 😄
  - Safe bet, since 23.5% of the tokens in the Brown Corpus are nouns
    - Second-largest, verbs, are 14.6%
  - Some words can have different parts-of-speech, and thus, different root forms, depending on context

```
>>> from nltk.stem import WordNetLemmatizer
>>> wnl = WordNetLemmatizer()
>>> wnl.lemmatize("operating")
'operating'
>>> wnl.lemmatize("operating", pos='v')
'operate'
>>> wnl.lemmatize("operating", pos='a')
'operating'
```

Using What We've Learned

- Tokenization, lemmatization, conditional frequencies, part-of-speech tagging
➡️ **Named Entity Recognition**

- Look at each token individually, and its
  - part-of-speech (not just **NNP**)
  - *n* words surrounding it ("sliding window")
  - The part-of-speech tags of those *n* words

Obama/NNP was/VBD born/VBN in/IN Honolulu/NNP ,/, Hawaii/NNP ,/, and/CC is/VBZ a/DT graduate/NN of/IN Columbia/NNP University/NNP and/CC Harvard/NNP Law/NNP School/NNP ./.

NER in Noisy Text

Date: Sat, 21 Jun 2003 03:44:38/TIME -0700/NUMBER (PDT)
From: Merrick Berg/PERSON <mmmmberg@yahoo.com>
To: lars@winds.gsfc.nasa.gov
Cc: Nilani/ORGANIZATION <anilani@cs.umd.edu>
Subject: computer

Hi Lars:
Call me around 4-5/NUMBER p.m.
if you would like to come over
this evening/TIME and configure your computer.I'll be
looking forward to see you tomorrow/DATE.
There are things that I don't understand in Treemap/LOCATION
that you may already know having spent so much time on it.
Maybe you could tell me
those things right away rather than me
trying to understand spending too much time
on them.

Other Incredibly Useful Things That I Don't Have Time to Talk About

- How to build better language models
- Measuring information
  - LSA, entropy, (point-wise) mutual information, t-score
- Document similarity
  - LSA (again), cosine similarity

- Johnson, Mark. "Natural Language Processing and Computational Linguistics: from Theory to Application". PowerPoint. *Macquarie University Department of Computing.* 2012. 13 July 2014. <http://web.science.mq.edu.au/~mjohnson/papers/CLandTopicModels.pdf>

- Manning, Christopher D. and Schütze, Hinrich. Foundations of Statistical Natural Language Processing. MIT Press. 1999.

- Jurafsky, Daniel and Martin, James H. Speech and Language Processing. Prentice Hall. 2008.

- Aluthgedara, Nilani. (2003). *Recognizing Sentence Boundaries and Boilerplate*. Bachelor Thesis. University of Maryland, College Park: U.S.

- Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology* (HLT '94). Association for Computational Linguistics, Stroudsburg, PA, USA, 114-119.

- Bird, Steven, Klein, Ewan, and Loper, Edward. Natural Language Processing with Python. O'Reilly Media. 2009.

- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005*.

- Johnson, Mark. "Natural Language Processing and Computational Linguistics: from Theory to Application". PowerPoint. *Macquarie University Department of Computing.* 2012. 13 July 2014. <http://web.science.mq.edu.au/~mjohnson/papers/CLandTopicModels.pdf>
- Manning, Christopher D. and Schütze, Hinrich. Foundations of Statistical Natural Language Processing. MIT Press. 1999.
- Jurafsky, Daniel and Martin, James H. Speech and Language Processing. Prentice Hall. 2008.
- Aluthgedara, Nilani. (2003). *Recognizing Sentence Boundaries and Boilerplate*. Bachelor Thesis. University of Maryland, College Park: U.S.
- Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology* (HLT '94). Association for Computational Linguistics, Stroudsburg, PA, USA, 114-119.
- Bird, Steven, Klein, Ewan, and Loper, Edward. Natural Language Processing with Python. O'Reilly Media. 2009.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005*.