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"
Introduction

Barack Hussein Obama III is the 44th and current President of the United States, in office since 2009.
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](https://via.placeholder.com/150)

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

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

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]]]]]] [. .]]

<table>
<thead>
<tr>
<th>Barack Hussein Obama II</th>
<th>PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>LOCATION</td>
</tr>
<tr>
<td>2009</td>
<td>DATE</td>
</tr>
</tbody>
</table>
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

---

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.

Syntactic Linguistic Units

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

<table>
<thead>
<tr>
<th></th>
<th>&quot;That&quot;, &quot;that&quot;, &quot;is&quot;, &quot;is&quot;, &quot;That&quot;, &quot;that&quot;, &quot;is&quot;, &quot;not&quot;, &quot;is&quot;, &quot;not&quot;, &quot;Is&quot;, &quot;that&quot;, &quot;it&quot;, &quot;It&quot;, &quot;is&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>15------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>&quot;That&quot;, &quot;that&quot;, &quot;is&quot;, &quot;is&quot;, &quot;.&quot;, &quot;That&quot;, &quot;that&quot;, &quot;is&quot;, &quot;not&quot;, &quot;,&quot;, &quot;is&quot;, &quot;not&quot;, &quot;.&quot;, &quot;Is&quot;, &quot;that&quot;, &quot;it&quot;, &quot;?&quot;, &quot;It&quot;, &quot;is&quot;, &quot;.&quot;</td>
</tr>
<tr>
<td>Tokens</td>
<td>20------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Types</td>
<td>&quot;that&quot;, &quot;is&quot;, &quot;,&quot;, &quot;.&quot;, &quot;not&quot;, &quot;it&quot;, &quot;?&quot;</td>
</tr>
<tr>
<td></td>
<td>7-------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th></th>
<th>run</th>
<th>running</th>
<th>ran</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stem</td>
<td>run</td>
<td>run</td>
<td>ran</td>
</tr>
<tr>
<td>Lemma</td>
<td>run</td>
<td>run</td>
<td>run</td>
</tr>
</tbody>
</table>
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
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

<table>
<thead>
<tr>
<th>bigrams</th>
<th>trigrams</th>
<th>4-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;you shall&quot;</td>
<td>&quot;you shall know&quot;</td>
<td>&quot;you shall know a&quot;</td>
</tr>
<tr>
<td>&quot;shall know&quot;</td>
<td>&quot;shall know a&quot;</td>
<td>&quot;shall know a word&quot;</td>
</tr>
<tr>
<td>&quot;know a&quot;</td>
<td>&quot;know a word&quot;</td>
<td>&quot;know a word by&quot;</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Modeling Language
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

```python
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

```python
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 (str(ngram), str(conditional_freq))
```

Remove non-content words
A Bigram Language Model of the Brown Corpus

```python
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)
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 (str(ngram), str(conditional_freq))
```

Remove non-content words

No bigrams of punctuation
A Bigram Language Model of the Brown Corpus

```python
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

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 😃

<table>
<thead>
<tr>
<th>Top 10 Most Frequent:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;united states&quot;</td>
</tr>
<tr>
<td>&quot;new york&quot;</td>
</tr>
<tr>
<td>&quot;per cent&quot;</td>
</tr>
<tr>
<td>&quot;years ago&quot;</td>
</tr>
<tr>
<td>&quot;rhode island&quot;</td>
</tr>
<tr>
<td>&quot;could see&quot;</td>
</tr>
<tr>
<td>&quot;last year&quot;</td>
</tr>
<tr>
<td>&quot;even though&quot;</td>
</tr>
<tr>
<td>&quot;high school&quot;</td>
</tr>
<tr>
<td>&quot;white house&quot;</td>
</tr>
</tbody>
</table>
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.

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"

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dogs</td>
<td>6</td>
<td>leash</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>dog</td>
<td>7</td>
<td>huskies</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>barking</td>
<td>8</td>
<td>wagging</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>barked</td>
<td>9</td>
<td>kennel</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>collie</td>
<td>10</td>
<td>manilak</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>
Querying an LSI for the 20 Most-Similar Terms

Demo: http://lsa.colorado.edu/

Corpus: "General Reading up to 1st Year College"

<table>
<thead>
<tr>
<th></th>
<th>&quot;dogs&quot;</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dogs</td>
<td>6</td>
<td>leash</td>
<td>11</td>
<td>snarling</td>
</tr>
<tr>
<td>2</td>
<td>dog</td>
<td>7</td>
<td>huskies</td>
<td>12</td>
<td>unshopped</td>
</tr>
<tr>
<td>3</td>
<td>barking</td>
<td>8</td>
<td>wagging</td>
<td>13</td>
<td>oogruk</td>
</tr>
<tr>
<td>4</td>
<td>barked</td>
<td>9</td>
<td>kennel</td>
<td>14</td>
<td>terrier</td>
</tr>
<tr>
<td>5</td>
<td>collie</td>
<td>10</td>
<td>manilak</td>
<td>15</td>
<td>puppies</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The top 20 similar terms to "dogs" are shown in the table above.
Querying an LSI for the 20 Most-Similar Terms

- **Demo:** [http://lsa.colorado.edu/](http://lsa.colorado.edu/)

- **Corpus:** "General Reading up to 1st Year College"

<table>
<thead>
<tr>
<th></th>
<th>&quot;dogs&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dogs</td>
</tr>
<tr>
<td>2</td>
<td>dog</td>
</tr>
<tr>
<td>3</td>
<td>barking</td>
</tr>
<tr>
<td>4</td>
<td>barked</td>
</tr>
<tr>
<td>5</td>
<td>collie</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>leash</td>
<td>11</td>
<td>snarling</td>
<td>12</td>
<td>unshopped</td>
</tr>
<tr>
<td>7</td>
<td>huskies</td>
<td>13</td>
<td>oogruk</td>
<td>14</td>
<td>terrier</td>
</tr>
<tr>
<td>8</td>
<td>wagging</td>
<td>15</td>
<td>puppies</td>
<td>19</td>
<td>stray</td>
</tr>
<tr>
<td>9</td>
<td>kennel</td>
<td>16</td>
<td>collies</td>
<td>17</td>
<td>pups</td>
</tr>
<tr>
<td>10</td>
<td>manilak</td>
<td>18</td>
<td>pups</td>
<td>20</td>
<td>mongrel</td>
</tr>
</tbody>
</table>
Querying an LSI for the 20 Most-Similar Terms

- Demo: [http://lsa.colorado.edu/](http://lsa.colorado.edu/)

- Corpus: "General Reading up to 1st Year College"

<table>
<thead>
<tr>
<th>&quot;dogs&quot;</th>
<th>1</th>
<th>dogs</th>
<th>6</th>
<th>leash</th>
<th>11</th>
<th>snarling</th>
<th>12</th>
<th>unshopped</th>
<th>13</th>
<th>oogruk</th>
<th>14</th>
<th>terrier</th>
<th>19</th>
<th>stray</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>dog</td>
<td>7</td>
<td>huskies</td>
<td>12</td>
<td>unshopped</td>
<td>13</td>
<td>oogruk</td>
<td>14</td>
<td>terrier</td>
<td>19</td>
<td>stray</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>barking</td>
<td>8</td>
<td>wagging</td>
<td>13</td>
<td>oogruk</td>
<td>14</td>
<td>terrier</td>
<td>19</td>
<td>stray</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>barked</td>
<td>9</td>
<td>kennel</td>
<td>14</td>
<td>terrier</td>
<td>19</td>
<td>stray</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>collie</td>
<td>10</td>
<td>manilak</td>
<td>15</td>
<td>puppies</td>
<td>20</td>
<td>mongrel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Querying an LSI for the 20 Most-Similar Terms

- **Demo:** [http://lsa.colorado.edu/](http://lsa.colorado.edu/)
- **Corpus:** "General Reading up to 1st Year College"

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;dogs&quot;</td>
<td>&quot;dogs&quot;</td>
<td>leash</td>
<td>snarling</td>
</tr>
<tr>
<td>1</td>
<td>dogs</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>dog</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>barking</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>barked</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>collie</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>leash</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>huskies</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>wagging</td>
<td>13</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>kennel</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>manilak</td>
<td>15</td>
<td>mongrel</td>
</tr>
<tr>
<td>11</td>
<td>snarling</td>
<td>16</td>
<td>pups</td>
</tr>
<tr>
<td>12</td>
<td>unshopped</td>
<td>17</td>
<td>puppy</td>
</tr>
<tr>
<td>13</td>
<td>oogruk</td>
<td>18</td>
<td>stray</td>
</tr>
<tr>
<td>14</td>
<td>terrier</td>
<td>19</td>
<td>mongrel</td>
</tr>
</tbody>
</table>
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 ./.
```

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRP</td>
<td>Personal Pronoun</td>
</tr>
<tr>
<td>MD</td>
<td>Modal Verb</td>
</tr>
<tr>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>VB</td>
<td>Verb, root form</td>
</tr>
<tr>
<td>IN</td>
<td>Preposition</td>
</tr>
<tr>
<td>EX</td>
<td>Existential <em>there</em></td>
</tr>
<tr>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
<tr>
<td>.</td>
<td>End of sentence</td>
</tr>
</tbody>
</table>
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

```python
>>> 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 <mrmmbberg@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
References


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


References


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


