NICF -Text Analytics

MODULE 5: GET THE TEXT DATA READY FOR ANALYSIS

Dr. Fan Zhenzhen
Institute of Systems Science
National University of Singapore
Email: zhenzhen@nus.edu.sg

Objectives

At the end of this module, you can:

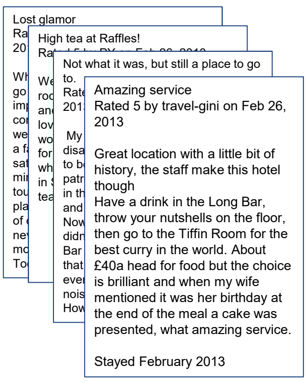
- Define the metadata and corpus to be imported into TA repository
- Identify preprocessing steps of text data typically required for TA tasks
- Develop term-document frequency matrix to enable lookup of text and documents within the corpus

Outline

- Overview of task
- Text preprocessing
- From text to word – tokenization, stemming, stopword removal
- Convert text to term-document frequency matrix (indexing)
- Other processing tasks (NLP)

The whole task here is…

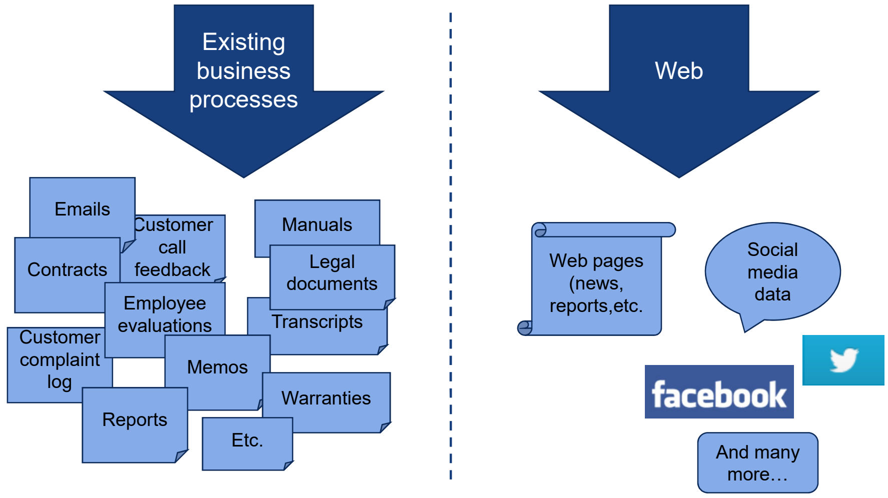
Documents



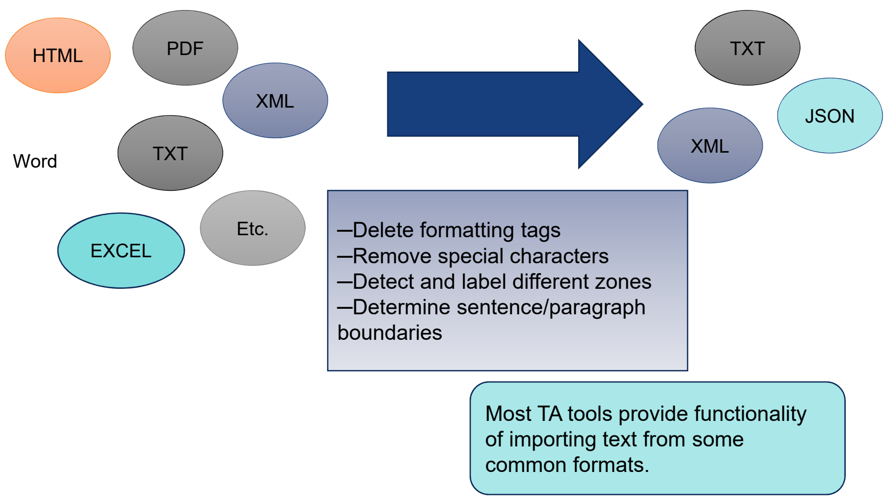
Term Document Matrix

	amazing	service	lost	glamour	disappoint	brilliant	super	expensive	noisy	…
Doc1	1	1	0	0	0	1	0	0	0	
Doc2	0	0	1	1	1	0	0	1	0	
Doc3	0	0	0	1	0	0	1	0	0	
Doc4	0	0	0	0	2	0	0	1	1	
…										

Sources of Text Data



File Preprocessing



In a pure text file…

Amazing service

Rated 5 by travel-gini on Feb 26, 2013

Great location with a little bit of history, the staff make this hotel though
Have a drink in the Long Bar, throw your nutshells on the floor, then go to the Tiffin Room for the best curry in the world. About £40a head for food but the choice is brilliant and when my wife mentioned it was her birthday at the end of the meal a cake was presented, what amazing service.

Stayed February 2013

…

XML as Standard Exchange Format

- A trend in industry and text-processing community
- With XML, we can insert tags onto a text to identify its parts.
  - Eg. *<DOC>*, *<SUBJECT>*, *<TOPIC>*, *<TEXT>*, etc.
  - Such tags are very useful as they allow selection/extraction of the parts to generate features for subsequent mining.
- Many word processors allow documents to be saved as XML format

What would an XML doc look like?

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<REVIEWS>
    <REVIEW>
        <TITLE>Amazing service</TITLE>
        <RATING>5</RATING>
        <DATE>26/02/2013</DATE>
        <BY>travel-gini</BY>
        <CONTENT>Great location with a little bit of history, the staff make this hotel though
    Have a drink in the Long Bar, throw your nutshells on the floor, then go to the Tiffin Room for the best
    curry in the world. About £40a head for food but the choice is brilliant and when my wife mentioned it
    was her birthday at the end of the meal a cake was presented, what amazing service. Stayed
    February 2013 </CONTENT>
    </REVIEW>
    …
</REVIEWS>
```


Or as JSON format

```
{
  "title": "Amazing service",
  "rating": 5,
  "date": "26/02/2013,
  "by": "travel-gini",
  "content": "Great location with a little bit of history, the staff make this
    hotel though Have a drink in the Long Bar, throw your nutshells on the
    floor, then go to the Tiffin Room for the best curry in the world. About
    £40a head for food but the choice is brilliant and when my wife
    mentioned it was her birthday at the end of the meal a cake was
    presented, what amazing service. Stayed February 2013"
}
```


From Text to Words

Tokenization

- To break a stream of characters into tokens

Great location with a little bit of history.

↓

Great	location	with	a	little	bit	of	history

- This is done by identifying token delimiters
  - Whitespace characters such as ***space, tab, newline***
  - Punctuation characters like ***( ) < > ! ? “ ”***
  - Other characters ***. , : - ‘ ’*** etc.
- It's known as unigram model – every token is a single word. There are also bigram, trigram models, where each token is composed of two/three words.

Tokenization Challenges

- It seems simple, but…
  - **. , :** between numbers are part of the number

    12.34 | 12,345 | 12:34

  - **.** can be part of an abbreviation or end of a sentence

    U.S.A. | Dr.

  - **'** can be a closing internal quote, indicate a possessive, or be part of another token

    My friend's | isn't

Lemmatization/Stemming

- A word may come in varied forms and therefore need to be converted into a standard form
  - **Inflectional** stemming (no change of POS)
  - **Derivational** Stemming (with change of POS)
  - Other normalisation (including case normalisation)

    U.S.A., US → USA

- Stemming can reduce the number of distinct features in a text corpus and increase the frequency of occurrence of some individual features.

Inflectional Stemming

- Inflections – grammatical variants of the same word
  - Plural form of nouns

    nutshells → nutshell

    classes → class

    stories → story

  - Verbs in different tense and aspect

    likes, liked, liking → like

- There are irregular forms and ambiguities!
  - "*corpus*" vs. "*corpora*", "*seek*" vs. "*sought*"
  - Is "*bore*" the present tense of "*bore*" or past tense of "*bear*"?

Derivational Stemming

- Derivation – forming new words from another word or stem by adding prefixes and/or suffixes

  production → produce

- Thus derivational stemming can change the syntactic category of a root

  reapply → apply

  unhappy → happy

- It may also cause a change of meaning

How much stemming should be done?

- An inflectional stemmer needs to be partly rule-based and partly dictionary-based.
- Derivational stemming is more aggressive and therefore can reduce the number of features in a corpus drastically. However meaning might be lost in the stemming process.

Too aggressive stemming can result in loss of meaning and non-legitimate words without the support of a dictionary.

```
[[6]]
battery life portability accessories style

[[7]]
   ability  store    music     ability  create playlists

[[8]]
portability capacity sound quality durability
```

```
[[6]]
batteri life portabl accessori style

[[7]]
   abil  store    music     abil  creat playlist

[[8]]
portabl capac sound qualiti durabl
```


Stemmers

- Some well-known stemming algorithms for English
  - Lovins Stemmer by Julie Beth Lovins, 1968
    - single pass, longest-match
    - removing the longest suffix, ensuring the remaining stem is at least 3 characters long
    - reforming the stem through recoding transformations
  - Porter Stemmer by Martin Porter, 1980
    - Widely used, with implementations in various languages available online (C, java, Perl, python, C#, VB, Javascript, Tcl, Ruby, etc.)
  - Snowball by Porter, a framework for writing Stemming algorithms

Stopword Removal

- Some words are extremely common. They appear in almost all documents and carry little meaning. They are of limited use in text analytics applications.
  - Functional words (conjunctions, prepositions, determiners, or pronouns) like *the, of, to, and, it,* etc.
  - A stopword list can be constructed to exclude them from analysis.
  - **Depending on the domain**, other words may need to be included in the stopword list.

Great | location | with | a | little | bit | of | history

Indexing

Indexing

- Many text mining applications are based on vector representation of documents (*term-document matrix* or *document-term matrix*) using “bag-of-words” approach

$$
\begin{array}{c|cccc}
 & T_1 & T_2 & \dots & T_t \\
\hline
D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\
D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\
: & : & : & & : \\
: & : & : & & : \\
D_n & w_{1n} & w_{2n} & \cdots & w_{tn}
\end{array}
$$

T: term
D: document
w: **weight** of the term

- Usually only content words (adjectives, adverbs, nouns, and verbs) are used as vector features.

Term Weighting

- Binary
    - 0 or 1, simply indicating whether a word has occurred in the document (but that’s not very helpful).
- Frequency-based
    - *term frequency*, the frequency of words in the document, which provides additional information that can be used to contrast with other documents.

	amazing	service	lost	glamour	disappoint	brilliant	super	expensive	noisy	...
Doc1	1	1	0	0	0	1	0	0	0	
Doc2	0	0	1	1	1	0	0	1	0	
Doc3	0	0	0	1	0	0	1	0	0	
Doc4	0	0	0	0	2	0	0	1	1	
...										

Frequent Word List

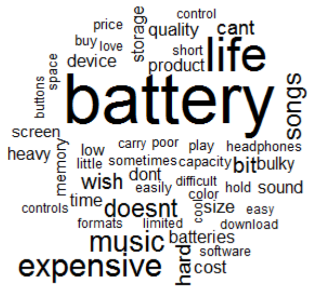
- With frequency-based TDM, a list of words and their frequencies in the corpus can be generated
  - *Global frequency* – how many times a word appears in the corpus
  - *Document frequency* – how many unique documents contain the word
- This list, sorted by frequency, can give us a rough idea of what the corpus is about.
- Word Cloud is a nice visualization of such information.

Word Cloud: another example

- Generated from http://worditout.com/word-cloud/make-a-new-one

Contrasting Two Groups

- "What do you like most…"
- "What do you like least…"



Other Weighting Methods

- Normalized frequency
  - To deal with varied document length, since a long document definitely has more occurrences of terms than a short document

$$normalized\_frequency = \frac{frequency\ of\ a\ term\ in\ a\ document}{total\ number\ of\ terms\ in\ the\ document}$$

- *tf-idf*
  - To modify the frequency of a word in a document by the perceived importance of the word(the *inverse document frequency)*, widely used in information retrieval
    - When a word appears in many documents, it's considered unimportant.
    - When the word is relatively unique and appears in few documents, it's important.

tf-idf Indexing

example :
5 occurrence in the document * (10000 document, 5 contain the term)

- *tf-idf* weighting :

$tf\text{-}idf_{t,d} = tf_{t,d} * idf_t$

- $tf_{t,d}$: <u>term frequency</u> – number of occurrences of term t in document d
- idf_t : <u>inverted document frequency</u> of term t

$$idf_t = \log \frac{N}{df_t}$$

N : the total number of documents in the corpus

df_t : the document frequency of term t, i.e., the number of documents that contain the term.

~common word will be close to 0

tf-idf Indexing – An Example

Note that in this example, stopwords and very common words are not removed, and terms are not re[...]

TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$

Query, Q: "gold silver truck"
D_1: "Shipment of gold damaged in a fire"
D_2: "Delivery of silver arrived in a silver truck"
D_3: "Shipment of gold arrived in a truck"
D = 3; IDF = log(D/df_i)

IDFi here is IDFt

	Counts, tf_i							Weights, $w_i = tf_i * IDF_i$			
Terms	**Q**	**D_1**	**D_2**	**D_3**	**df_i**	**D/df_i**	**IDF_i**	**Q**	**D_1**	**D_2**	**D_3**
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761

http://www.miislita.com/term-vector/term-vector-3.html

Note that in this example, stopwords and very common words are not removed, and terms are not reduced

The higher the TF*IDF score (weight), the rarer the term and vice versa.

Alternative Representation of TDM

- The resulting term document matrix is expected to have most of the values to be zero, since typically a document will only contain a small subset of the vocabulary in a corpus

```
<<DocumentTermMatrix (documents: 1000, terms: 17887)>>
Non-/sparse entries: 92858/17794142
Sparsity           : 99%
Maximal term length: 56
Weighting          : term frequency (tf)
```

- It saves memory to store the matrix as a set of sparse vectors, where a row is represented by a list of pairs, (ColumnNumber, Value)

Matrix

0	5	2	0
4	0	0	0
3	1	0	6

⇨

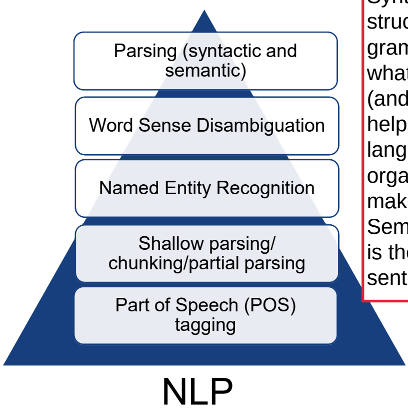
Sparse Vectors

(2, 5) (3, 2)
(1, 4)
(1, 3) (2, 1) (4, 6)

Natural Language Processing Tasks

Natural Language Processing Tasks

- To extract more sophisticated features, additional linguistic analyses of the text is needed.



Syntax is the study of sentence structure and the rules of grammar. While people can do what they want with language (and many often do), syntax helps common users of a language understand how to organize words so that they make the most sense. Semantics, on the other hand, is the study of the meaning of sentences.

WSD is basically solution to the ambiguity which arises due to different meaning of words in different context. For example, consider the two sentences. “The bank will not be accepting cash on Saturdays. ” “The river overflowed the bank.”

POS Tagging

- To determine POS or grammatical category of a term, using rule-based or stochastic approach
  - Nouns, verbs, adjectives, adverbs, pronouns, determiners, prepositions, conjunctions, etc.
  - Dictionary with word-POS correspondence is needed
- Challenge – POS disambiguation (words with >1 POS)
  - E.g. “*book*” can be a noun (“*my **book***”) or a verb (“*to **book** a room*”)

IN/ About CD/ six CC/ and DT/ a JJ/ half NNS/ hours RB/ later ,/ , NNP/ Mr. NNP/ Armstrong VBD/ opened DT/ the NN/ landing NN/ craft POS/ 's NN/ hatch ,/ , VBD/ stepped RB/ slowly IN/ down DT/ the NN/ ladder CC/ and VBD/ declared IN/ as PRP/ he VBD/ planted DT/ the JJ/ first NN/ human NN/ footprint IN/ on DT/ the NN/ lunar NN/ crust :/ : ``/ " DT/ That VBZ/ 's CD/ one JJ/ small NN/ step IN/ for NN/ man ,/ , CD/ one JJ/ giant NN/ leap IN/ for NN/ mankind ./ . "/ "

Generated by UIUC POS Tagger

CC coordinating conjunction.
CD cardinal digit.
DT determiner.
EX existential there (like: “there is” … ...
FW foreign word.
IN preposition/subordinating conjunction.
JJ adjective 'big'
JJR adjective, comparative 'bigger'

POS Taggers

- Rule-based - e.g. Brill's tagger by Eric Brill
  - Error-driven transformation-based tagger
  - Initially assign the most frequent tag to each word, based on dictionary and morphological rules
  - Contextual rules are then applied repeatedly to correct any errors
- Stochastic taggers – e.g. CLAWS, Viterbi, Baum-Welch, etc.
  - based on Hidden Markov Models (HMMs) and n-gram probabilities
  - Manually tagged corpus is needed to estimate probabilities
- Many machine learning methods have also been applied
- Stanford's Statistical NLP website lists many free taggers

Morphology is the study of words. Morphemes are the minimal units of words that have a meaning and cannot be subdivided further. . An example of a free morpheme is "bad", and an example of a bound morpheme is "ly." It is bound because although it has meaning, it cannot stand alone.

Shallow Parsing / Chunking

- To identify phrases in a text (noun phrases, verb phrases, and prepositional phrases, etc.)
- Largely stochastic techniques based on probabilities derived from an annotated corpus
- Faster, more robust than full parsing

[NP About six and a half hours] [ADVP later] , [NP Mr. Armstrong] [VP opened] [NP the landing craft] [NP 's hatch] , [VP stepped] [ADVP slowly] [PP down] [NP the ladder] and [VP declared] [SBAR as] [NP he] [VP planted] [NP the first human footprint] [PP on] [NP the lunar crust] : "[NP That] [VP 's] [NP one small step] [PP for] [NP man] , [NP one giant leap] [PP for] [NP mankind] ."

Generated by UIUC chunker

An example of a prepositional phrase is, "With a reusable tote in hand, Matthew walked to the farmer's market." Every prepositional phrase is a series of words consisting of a preposition and its object. In the example above, "with" is the preposition and "reusable tote" is the object.

Shallow parsing (also chunking or light parsing) is an analysis of a sentence which first identifies constituent parts of sentences (nouns, verbs, adjectives, etc.) and then links them to higher order units that have discrete grammatical meanings (noun groups or phrases, verb groups, etc.).

Name Entity Recognition

- Recognition of particular types of proper noun phrases, specifically *persons*, *organizations*, *locations*, and sometimes *money*, *dates*, *times*, and *percentages*.
- Very useful in text analytics applications, by turning verbose text data into a more compact structural form
- **More details in another module**

[**LOC Houston**] , Monday, July 21 -- Men have landed and walked on the moon. Two [**MISC Americans**] , astronauts of [**ORG Apollo**] 11, steered their fragile four-legged lunar module safely and smoothly to the historic landing yesterday at 4:17:40 P.M., Eastern daylight time. [**PER Neil A. Armstrong**] , the 38-year-old civilian commander, radioed to earth and the mission control room here: "[**LOC Houston**] , [**ORG Tranquility Base**] here; the Eagle has landed."

Generated by UIUC NER system

Word Sense Disambiguation

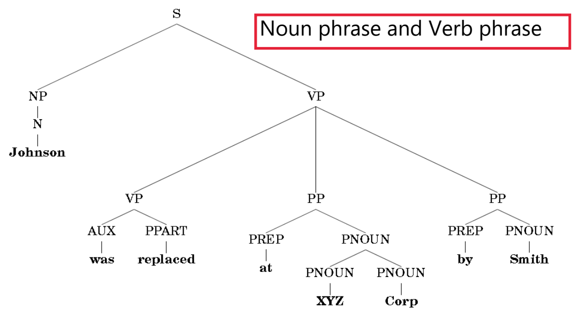
- Words are also ambiguous as to their meaning or reference
  - E.g. *table*: 1. a piece of furniture with a flat top supported by legs
    2. A list of numbers, facts, or information arranged in rows across and down a page
- Disambiguation of meanings in context has not been well solved, partly due to the lack of corpus of disambiguated text to serve as training corpus for machine learning algorithms
- Usually not applied in a typical text analytics application

Parsing

- Or *Syntactic Analysis*, the more sophisticated kind of text processing
- To produce a full parse of a sentence, typically as a tree, with syntactic functions of each word (e.g. subject, object, etc.)

Johnson was replaced at XYZ Corp by Smith

Noun phrase and Verb phrase



Syntactic analysis, also referred to as syntax analysis or parsing, is the process of analyzing natural language with the rules of a formal grammar.

Parsing

- Tree can be represented in another way:

Tagging

```
John/NNP  was/VBD  replaced/VBN  at/IN  XYZ/NNP  Corp/NNP  by/IN  Smith/NNP  ./.
```

Parse

```
(ROOT
  (S
    (NP (NNP John))
    (VP (VBD was)
      (VP (VBN replaced)
        (PP (IN at)
          (NP (NNP XYZ) (NNP Corp)))
        (PP (IN by)
          (NP (NNP Smith)))))
    (. .)))
```

From Stanford Parser

Typed dependencies, collapsed

```
nsubjpass(replaced-3, John-1)
auxpass(replaced-3, was-2)
root(ROOT-0, replaced-3)
nn(Corp-6, XYZ-5)
prep_at(replaced-3, Corp-6)
agent(replaced-3, Smith-8)
```

- Comparatively expensive process, but can provide information that shallow parsing can not provide.

From Syntax to Semantics

- Semantic analysis can be applied on top of parsing result to help identify the right entity for the text mining task.

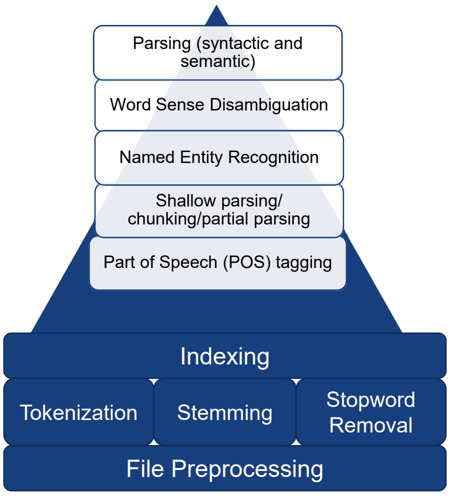
	SRL	Charniak
John	old thing [A1]	(S1 (S (NP (NNP John))
was		(VP (AUX was)
replaced	V: replace	(VP (VBN replaced)
at	location [AM-LOC]	(PP (IN at)
XYZ		(NP (NNP XYZ)
Corp		(NNP Corp)))
by	replacer [A0]	(PP (IN by)
Smith		(NP (NNP Smith)))))
.		(. .)))

Generated by UIUC Semantic Role Labeling system

Challenges in Parsing

- Robustness – graceful degradation
  - The input may not conform to what is normally expected
  - Ill-formed input or lack of coverage of grammars
  - To recover as much meaningful information as possible
- Disambiguation
  - Ambiguity accumulated from earlier steps can result in combinatorial increase of possible parses
  - Return the *n* best analyses, if not one, to the next level of processing
- Efficiency
  - Theoretical time complexity of most formalisms are polynomial

Summary



Reference & Resources

- Weiss, Indurkhya, & Zhang. Chapter 2 “From Textual Information to Numerical Vectors”, *Fundamentals of Predictive Text Mining*, Springer, 2010.
- List of online word cloud generators
  - http://www.techlearning.com/default.aspx?tabid=67&entryid=364
- UIUC POS Tagger, Chunker, etc.
  - http://cogcomp.cs.illinois.edu/page/demos
- NLP resources: http://nlp.stanford.edu/links/statnlp.html