CS544: Classification Algorithms

February 7, 2012

Zornitsa Kozareva
USC/ISI
Marina del Rey, CA
kozareva@isi.edu
www.isi.edu/~kozareva

Today

• Continuing with ML

• Binary classification
  – Perceptron

• Features
Binary Classification (Example)

• Spam filtering
  – spam vs. not spam

• Message classification
  – urgent vs. not urgent

• Sentiment classification
  – positive vs. negative

• Sometime it can be convenient to treat a multi-way problem like a binary one: one class versus all the others, for all classes

Binary Classification

• Given: some data items that belong to a positive (+1 ⬤) or a negative (-1 ⬤) class

• Task: train a classifier to predict the class for a new data item

• Geometrically: find a separator
Linearly Separable Data

Non linearly Separable Data
Linear vs. Non Linear Algorithms

- **Linearly separable data**: if all the data points can be correctly classified by a linear decision boundary

- **Linear**
  - Advantages: simpler, less parameters
  - Disadvantages: high dimensional data is usually not linearly separable
  - Example: Perceptron

- **Non linear**
  - Advantages: more accurate
  - Disadvantages: more complicated, more parameters
  - Example: Kernel Methods
Perceptron

\[ w_{ij} \leftarrow w_{ij} + \eta(t_j - y_j)x_i \]

where \( w_{ij} \) is the weighted connection between nodes \( i \) and \( j \), \( \eta \) is the learning rate, \( t_j \) is the target vector with "correct" answers, \( y_j \) is the output vector, \( x_i \) is the vector \( x \) with elements \( (x_1, x_2, \ldots, x_n) \), \( n \) is the number of dimensions, \( g(.) \) is the activation function, \( y_i \) is the output vector, \( t \) is the target vector containing the "correct" answers that the algorithm is learning about, and \( E \) is a function that computes the inaccuracies of the outputs \( y \) and targets \( t \).

Some Terminology

- **Inputs**: Vector \( x \) with elements \( (x_1, x_2, \ldots, x_n) \), \( n \) is number of dimensions
- **Weights**: \( w_{ij} \) are the weighted connection between nodes \( i \) and \( j \)
- **Outputs**: output vector \( y \)
- **Targets**: target vector \( t \) containing the "correct" answers that the algorithm is learning about
- **Activation Function**: \( g(.) \) describes the firing of a neuron as a response to the weighted inputs
- **Error**: \( E \) is a function that computes the inaccuracies of the outputs \( y \) and targets \( t \)
Perceptron Algorithm

- **Initialization**
  - set all weights \( w_{ij} \) to small (positive and negative) random numbers

- **Training**
  - for \( T \) iterations:
    - for each input vector:
      - compute the activation of each neuron \( j \) using activation function \( g \)
        \[
        y_i = g(\sum_{j=0}^{m} w_{ij} x_i) = \begin{cases} 
        1 & \text{if } w_{ij} x_i > 0 \\
        0 & \text{if } w_{ij} x_i \leq 0 
        \end{cases}
        \]
      - update each of the weights individually using:
        \[
        w_{ij} \leftarrow w_{ij} + \eta (t_j - y_j)x_i
        \]

Walkthrough Example

**Given:**

<table>
<thead>
<tr>
<th>ln1</th>
<th>ln2</th>
<th>t (target)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Initialize:**

\[
\begin{align*}
  w_0 &= -0.05 \\
  w_1 &= -0.02 \\
  w_2 &= 0.02 \\
  \eta &= 0.25
\end{align*}
\]

\[
\begin{align*}
  w_{ij} \leftarrow w_{ij} + \eta (t_j - y_j)x_i
\end{align*}
\]

\[
\begin{align*}
  y_i = g(\sum_{j=0}^{m} w_{ij} x_i) = \begin{cases} 
    1 & \text{if } w_{ij} x_i > 0 \\
    0 & \text{if } w_{ij} x_i \leq 0 
    \end{cases}
\end{align*}
\]
Non Linear Problem

- Kernel methods
- A family of non-linear algorithms
- Transform the non linear problem in a linear one (in a different feature space)
- Use linear algorithms to solve the linear problem in the new space

Basic Principle Kernel Methods

\[ \Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D \ (D >> d) \]

\[ w^T \Phi(x) + b = 0 \]

\[ f(x) = \text{sign}(w_1 x^2 + w_2 z^2 + w_3 x z + b) \]
Useful Books

★ Machine Learning: An Algorithmic Perspective, Stephen Marsland
http://www.amazon.com/Machine-Learning-Algorithmic-Perspective-Recognition/dp/1420067184/ref=sr_1_17
s=books&ie=UTF8&qid=1328600101&sr=1-1

• Pattern Recognition and Machine Learning, Christopher Bishop
http://research.microsoft.com/en-us/um/people/cmbishop/prml/

Note, these books are my personal suggestion for getting to know more about the topic, they are not obligatory for the course and it is not necessary for you to purchase them.
What is a Features?

• A feature is a quantitative measurement of some data element
  – numerical
  – binary
  – categorical
  – dependency tree

Feature Selection

• If the algorithm cannot handle all possible features
  – language identification for 100 languages using all words
  – text classification using n-grams

• Good features result in higher accuracy
  – sometimes keeping unreliable features can be helpful, but we need to weight the features (say, bad features will be 0)
Feature Selection

• Bad features: best to remove
  – too frequent
    • mostly function words (prepositions, determiners)
  – infrequent
    • unlikely to be seen again
    • co-occurrence with a class can be due to chance
  – uniform across all classes
• Good Features: should be kept
  – co-occur with a particular class
  – do not co-occur with other classes
• The rest: good to keep

Feature Selection

• Feature selection reduces the number of features
• This can be done using feature:
  – elimination
  – weighting
  – normalization
Features for Named Entity Recognition

Features for NER

Adam
Smith
Works
for
IBM
in
London
.

• **Contextual**
  • current word $W_0$
  • words around $W_0$ in [-3, ..., +3] window
Features for NER

- **Contextual**
  - current word $W_0$
  - words around $W_0$ in [-3,…,+3] window

Adam Smith Works for IBM in London

- fp

Adam, null, null, null, Smith, works, for Smith, Adam, null, null, works, for, IBM

- fp, London, in, IBM, null, null, null

- Contextual
  - current word $W_0$
  - words around $W_0$ in [-3,…,+3] window
Features for NER

Adam Smith Works for IBM in London .

fp, London, in, IBM, null, null, null, 0, 0

• Orthographic
  • initial-caps
  • all-caps

Features for NER

• Orthographic (binary and not mutually exclusive)
  • initial-caps
  • roman-number
  • acronym
  • single-char
  • all-caps
  • contains-dots
  • lonely-initial
  • functional-word*
  • all-digits
  • contains-hyphen
  • punctuation-mark
  • URL

• Word-Type Patterns:
  • functional capitalized
  • lowercased punctuation mark
  • quote
  • other

• Left Predictions
  • the tag predicted in the current classification for W-3, W-2, W-1

*functional-word is preposition, conjunction, article
Features for NER

• **Part-of-speech tag** (when available)
  - look mainly for proper names

• **Trigger words**
  - for person (*Mr, Miss, Dr, PhD*)
  - for location (*city, street*)
  - for organization (*Ltd., Co.*)

• **Gazetteers (big dictionaries of names)**
  - geographical
  - first name
  - surname
  - company names

Features for NER

• Length in words of the entity being classified

• **For each classs**
  - whole NE is in gazetteer
  - any component of the NE appears in gazetteer

• **Suffixes** (length 1 to 4)
  - each component of the NE
  - whole NE
Features for NER

- Previous word is an article
- Previous word is a noun
- Need more ideas on features, look at:

External Resources
Gazetteer Collection Method 1

- Yago contains over 2 million entities (like persons, organizations, cities among others)

- Download Yago from:
  

- Extract from the relevant relations all named entities
  Ex.
  - $X$ born in $Y$, where $X$ is a person and $Y$ is a location
  - $X$ works for $Y$, where $X$ is a person and $Y$ is a person or an organization

---

Gazetteer Collection Method 2

Madonna (entertainer)

From Wikipedia, the free encyclopedia

Madonna (born Madonna Louise Ciccone; August 16, 1958) is an American recording artist, actress and entrepreneur. Born in Bay City, Michigan, and raised in Rochester Hills, Michigan, she moved to New York City in 1977 to pursue a career in dance. After performing as a member of the pop musical groups Breakfast Club and Emmy, she released her self-titled debut album, Madonna, in 1983 on Sire Records. A series of hit singles from her next studio albums, Like a Virgin (1984) and True Blue (1986), gained her global recognition. They established her as a pop icon, pushing the boundaries of lyrical content in mainstream popular music and imagery in her music videos, which became a feature on MTV. Her recognition was augmented by the film Desperately Seeking Susan (1985) which widely became seen as a Madonna vehicle, despite her not playing the lead. Expanding on the use of religious imagery with Like a Prayer (1989), Madonna received positive critical reception for her diverse musical productions, while at the same time she was criticized by religious conservatives and the Vatican. In 1989, Madonna founded the Maverick label, a joint venture between herself and Time Warner. The same year, she expanded the use of sexually explicit material in her work, beginning with the release of the studio album Erotica, followed by the positivity of the coffee table book Sex, and starring in the erotic thriller Body of Evidence, all of which received negative responses from conservatives and liberal alike.

In 1996, Madonna played the starring role in the film Evita, for which she won a Golden Globe Award for Best Actress in a Motion Picture Musical or Comedy. Madonna's seventh studio album, Ray of Light (1998), became one of her most critically acclaimed, recognized for its lyrical depth. During the 2000s, Madonna released four studio albums—namely Music (2000), American Life (2003), Confessions on a Dance Floor (2005) and Hard Candy (2008)—all of which debuted at number one on the Billboard 200. In 2008, Madonna signed an unprecedented $120 million dollar contract with Live Nation in 2008.

According to the International Federation of the Phonographic Industry, Madonna has sold more than 200 million albums worldwide. She is ranked by the Recording Industry Association of America as the best-selling female rock artist of the 20th century, and the second-top selling female artist in the United States, behind Barbra Streisand, with 16 million certified albums. Billboard Hot 100 All-Time Top Artists, making her the most successful solo artist in the history of the chart. She was also inducted into the Rock and Roll Hall of Fame in the same year. Considered to be one of the most influential women in contemporary music, Madonna has been known to continually reinventing both her music and image, and for retaining a standard of autonomy within the recording industry. She is recognized as an influence among numerous music artists.
Gazetteer Collection Method 2

- Step 1: Check if identified NE exists in Wikipedia
- Step 2: Extract the first 2-3 sentences
- Step 3: Pull the nouns matching the expression
  \[ X \text{ is } Y, Z \]
  \[ X \text{ is } Y \text{ and } Z \]
- Step 4: Extract the information from the infobox
- Step 5: Verify in WordNet whether the found concepts are hyponyms of person, location, organization

(Madonna is an artist, actress)

Gazetteer Collection Method 3

- contains structured information from Wikipedia

<table>
<thead>
<tr>
<th>Class</th>
<th>Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place</td>
<td>462,000</td>
</tr>
<tr>
<td>Person</td>
<td>364,000</td>
</tr>
<tr>
<td>Organization</td>
<td>148,000</td>
</tr>
<tr>
<td>Resource (overall)</td>
<td>1,667,000</td>
</tr>
</tbody>
</table>

SPARQL results:

http://wiki.dbpedia.org/Datasets#h18-11
Other Gazetteer Sources

- The 2000 U.S. Census data

- Freebase (I think they made it payed)
  [http://www.freebase.com/schema/people](http://www.freebase.com/schema/people)

- Linked Data Sets
  [http://esw.w3.org/DataSetRDFDumps](http://esw.w3.org/DataSetRDFDumps)

...
Given Training Data Examples

<table>
<thead>
<tr>
<th>example</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Avatar" /></td>
<td>PERSON</td>
</tr>
<tr>
<td><img src="image" alt="Microsoft Logo" /></td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td><img src="image" alt="Avatar" /></td>
<td>PERSON</td>
</tr>
<tr>
<td><img src="image" alt="Map" /></td>
<td>LOCATION</td>
</tr>
<tr>
<td><img src="image" alt="Organization Logo" /></td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td><img src="image" alt="Map" /></td>
<td>LOCATION</td>
</tr>
<tr>
<td><img src="image" alt="Logo" /></td>
<td>OTHER</td>
</tr>
</tbody>
</table>
Choose a machine learning classifier from Weka

<table>
<thead>
<tr>
<th>example</th>
<th>Cap.</th>
<th>inDicPer</th>
<th>inDicOrg</th>
<th>inDicLoc</th>
<th>NP</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>PERSON</td>
</tr>
<tr>
<td>Microsoft</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>PERSON</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>LOCATION</td>
</tr>
<tr>
<td>ASI</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>LOCATION</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>OTHER</td>
</tr>
</tbody>
</table>

nxm matrix, where n is number of examples, m is number of features+class label
Given Test Data examples

<table>
<thead>
<tr>
<th>example</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
</tr>
<tr>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

Note that the class is unknown for the examples of the test data.

<table>
<thead>
<tr>
<th>example</th>
<th>Cap.</th>
<th>inDicPer</th>
<th>inDicOrg</th>
<th>inDicLoc</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>IBM</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
WEKA
Waikato Environment for Knowledge Analysis
Weka: Data Mining Software

- Collection of machine learning algorithms
  - open-source package written in Java
- Used for research, education and application
- Main features:
  - data pre-processing tools
  - learning algorithms
  - evaluation methods
  - graphical inference
  - environment for comparing learning algorithms

Weka: Data Mining Software

- Classification algorithms:
  - decision trees, linear classifiers, SVM, Naive-bayes, kNN
- Prediction algorithms:
  - regression (linear/SVM), perceptron
- Meta-algorithms:
  - bagging, boosting (AdaBoost)

among others
Getting Started

• Install Weka software (on Linux):
  – Download link:
    • http://prdownloads.sourceforge.net/weka/weka-3-6-2.zip
    • Unzip the software
  – Requirement:  Java 1.5 (or higher)
  – Invoke Weka command:
    • java -cp weka.jar <weka-command>

Weka GUI Chooser

```
java -Xmx1000M -jar weka.jar
```
Data file format (.arff)

@relation english_named_entity

@attribute position numeric
@attribute pos_tag { NN, NP, VB, DT}
@attribute word_length numeric
@attribute in_gazetteer { no, yes}
@attribute class { PER, LOC, ORG, MISC}

@data
3,DT,3,no,ORG
4,NP,10,yes,ORG
15,NP,6,yes,PER
7, NN,12,?,MISC
...

Other attribute types:
• String
• Date

The Preprocessing Tab
- Classification
- Preprocessing
- Statistical attribute selection
- Filter selection
- Manual attribute selection
- List of attributes (last: class variable)
- Statistics about the values of the selected attribute
- Frequency and categories for the selected attribute

Slide adapted from Marti Hearst
Choice of classifier

The attribute whose value is to be predicted from the values of the remaining ones. Default is the last attribute.

Cross-validation: split the data into e.g. 10 folds and 10 times train on 9 folds and test on the remaining one.

Choosing a classifier
all other numbers can be obtained from it

different/easy class

accuracy

Running on Test Set
WEKA
Command Line

Weka specifications

• Train classifier on training data and output model
  • java -cp weka.jar <classifier-function> -t <train-file> -d <trained-model>

• Run trained classifier model on test data
  • java -cp weka.jar <classifier-function> -T <test-file> -l <trained-model>

• Specifying parameters:
  -t : training file (.arff)
  -T : test file (.arff)
  -d : output filename (trained classifier model)
  -l : input model (for testing)
  -K : number of nearest neighbors for kNN algorithm
  -h : help (check out other parameter options, etc.)
Example: kNN in Weka

• Train a classifier using 2NN algorithm
  
  ```
  java -cp weka.jar
  weka.classifiers.lazy.IBk
  -t data/weather.arff
  -K 2
  -d model.2nn
  ```

• Run the trained classifier on test data
  
  ```
  java -cp weka.jar
  weka.classifiers.lazy.IBk
  -T data/weather.arff
  -l model.2nn
  ```

Sample Weka output

```plaintext
=== Error on test data ===
Correctly Classified Instances  13  92.8571 %
Incorrectly Classified Instances  1  7.1429 %
Kappa statistic  0.8372
Mean absolute error  0.1333
Root mean squared error  0.2333
Total Number of Instances  14
```
More detailed output

- Classification labels for each instance (use “–p 1” option)
  - java -cp weka.jar weka.classifiers.lazy.lbk -T data/weather.arff -I model.2nn -p 1

--- Predictions on test data ---

<table>
<thead>
<tr>
<th>Inst#</th>
<th>Actual</th>
<th>Predicted</th>
<th>Error</th>
<th>Prediction (outlook)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>sunny</td>
</tr>
<tr>
<td>2</td>
<td>2:no</td>
<td>1:yes + 0.5</td>
<td>sunny</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>4</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>5</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>6</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>7</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>8</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>sunny</td>
</tr>
<tr>
<td>9</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.5</td>
<td>sunny</td>
</tr>
<tr>
<td>10</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>11</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.5</td>
<td>sunny</td>
</tr>
<tr>
<td>12</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>13</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>14</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>rainy</td>
</tr>
</tbody>
</table>

Weka classification functions

- kNN: `weka.classifiers.lazy.lbk`
- Decision trees: `weka.classifiers.trees.J48`
- Naïve Bayes: `weka.classifiers.bayes.NaiveBayes`
- AdaBoost: `weka.classifiers.meta.AdaBoostM1`
Additional Information

• General documentation:
  http://www.cs.waikato.ac.nz/ml/weka/
  http://prdownloads.sourceforge.net/weka/weka.ppt

• Command line doc:
  http://weka.wikispaces.com/Primer

Assignment 1
Named Entity Challenge
• **Given**: a train and development set of English sentences tagged with four named entity classes:
  – PER (people)
  – ORG (organization)
  – LOC (location)
  – MISC (miscellaneous)

• **Your objective is**: to develop a machine learning NE system, which when given a new previously unseen text (i.e. test set) will identify and classify the named entities correctly.

---

**Data Description**

• The data consists of three columns separated by a single space. Each word has been put on a separate line and there is an empty line after each sentence.

- B-TYPE means the word is the beginning of a phrase of type TYPE
- O means the word is not part of a phrase

*Make sure to preserve the empty lines in the output of the test data*
Timeline

<table>
<thead>
<tr>
<th>Event</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train &amp; Development data</td>
<td>February 9th 2012</td>
</tr>
<tr>
<td>Test data</td>
<td>February 23rd 2012</td>
</tr>
<tr>
<td>Result submission deadline</td>
<td>February 25th 2012 (11:59 pm GMT)</td>
</tr>
<tr>
<td>later submissions will not be accepted</td>
<td></td>
</tr>
<tr>
<td>Presentation &amp; paper description deadline</td>
<td>February 25th 2012</td>
</tr>
</tbody>
</table>

Submit

- The source code for the feature generation *(make sure it will run under Linux)*
- The official train and test feature files used in the final run, together with the final output of your system for the test data
- Additionally generated resources (if any)
- Write 1-2 page brief description of your approach explaining:
  - used NLP tools
  - designed features
  - employed machine learning algorithm & motivation
Evaluation is based on

• ranking of your system against the rest
• designed features
  – novel, previously unknown features will be favored
  – system’s pre or post processing
• generated resources
  – size, methods and sources for gazetteer extraction
  – trigger lists
• quality of the paper description
  – structure
  – use of literature
  – error analysis

Generate Your Own Resources

• Extract gazetteers from Wikipedia
  – People (singers, teachers, mathematicians etc.)
  – Locations (cities, countries)
  – Organizations (universities, IT companies etc.)

• Extract trigger words from WordNet
  – look for hyponyms of person, location, organization

• Extract and rank the patterns in which the NEs occurred in the train and development data. Show what percentages of these were found in the final test data.

• Extract lists of verbs found next to the NEs. Do you find any similarity/regularity of the verbs associated with each one of the NE categories?
What must I do ...

- Use the train and development data to design and tune your NE system
- Decide on the features you would like to incorporate in your NE system
- Choose a machine learning classifier from Weka
  - Intro by Marti Hearst
    [http://courses.ischool.berkeley.edu/i256/f06/lectures/lecture16.ppt](http://courses.ischool.berkeley.edu/i256/f06/lectures/lecture16.ppt)
- or use other good toolkits like
  - CRF++
  - SVM light, LibSVM
    etc.
  
    This is a big assignment, try to start early!

What you must not do ...

- **Use existing named entity system(s) or library either as a feature generator, output generator etc.**

- If you do, then you will have to run your system for another language say Spanish 😊
Available Resources

- **WordNet** [http://wordnet.princeton.edu/](http://wordnet.princeton.edu/)
- **Part-of-speech taggers**
  - TreeTagger  
    [http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html](http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html)
  - Stanford PoS Tagger  
- **NP chunker**  
  - [http://www.dcs.shef.ac.uk/~mark/index.html](http://www.dcs.shef.ac.uk/~mark/index.html)
  - [http://www.dcs.shef.ac.uk/~mark/phd/software/chunker.html](http://www.dcs.shef.ac.uk/~mark/phd/software/chunker.html)
- **Parser**  
  - Stanford Parser  
- **Other**  

Good Luck!