Introduction to Natural Language Processing
This week & next week:
Classification
Sentiment Lexicons
Kendall review of HW 2
Next two weeks

- We are going to practice what we learned for processing texts with POS, ngrams etc, and use it in classification.

- We will classify different kinds of language, repeatedly using the same techniques.

- Solidify what we have learned so far.

- Chapter 5. Categorizing and Tagging Words
  - READ THE SECTION ABOUT DICT

- Chapter 6. Learning to Classify Text
  - READ THE CHAPTER
NLP PIPELINE: Bringing together

WORDS (MORPHOLOGY)
- Words, stemmed words

PATTERNS OF WORDS (DISTRIBUTIONAL ANALYSIS, LEXICAL SEMANTICS)
- Bigrams, Word categories

PHRASES AND SENTENCES (SYNTAX)
- POS, regexp

CLASSIFYING TEXTS (SEMANTICS)

SENTENCE MEANING (SEMANTICS)

DISCOURSE MEANING NARRATIVE STRUCTURES (SEMANTICS, PRAGMATICS, DISCOURSE)
Detecting patterns is core to NLP

- Learning a classifier model is one way to detect patterns (works best when combined with actually looking at the data yourself)

- How can we identify particular features of language data that are salient for classifying it?
  - ‘ed’ usually marks a past tense verb
  - terms like ‘Oh really’ often occur in sarcastic utterances

- How can we automatically construct models of language that can be used to perform language processing?

- What can we learn about language from these models?
Patterns are Key

- Same techniques used for images: what patterns distinguish drinking vessels?
- Is this utterance sarcastic?
- Is this movie review ‘thumbs up or thumbs down’?
Tweets from our work on sarcasm

- This totally topped off my week :'
- Electric picnic has a fantastic line up this year #wow
- Football and hockey are the only two things I'm looking forward to this school year ??
- Take a long time to reply and I'll take TWICE as long :)
- so happy work has started again
- I wish this feeling for you would just disappear </3
- I really just love my class. You are all too smart. #UhItsHighSchool
- That awkward moment when Taylor Swift does get back with her ex o.O
- My top lip is going to swell up right before we go back to school ): #Attractive
- Awkward eye contact is just fantastic
- What beautiful passport photos I just took #vom
- Feels great when I can't sleep, especially when all I want to do is talk to you.
- im gonna loveeee waking up at 5am everyday for schoollll
- Haha wow it's amazing how some seniors can leave after fourth period #jealous ??
- Love getting home from work knowing that in less than 8hours you're getting up to go back there again.

Which is which?
<table>
<thead>
<tr>
<th>Post Pair</th>
</tr>
</thead>
</table>
| **Q1:** Not only have I undercut your snails.  
**R1:** Oh No, everyone! Our Mollusca has been undercut! Whatever shall we do? :xbanghead
emoticon-rolleys |
| **Q3:** Did you know that in Siberia they are HAPPY that it's getting warmer up there?  
**Q3:** Actually global warming causes it to be hotter in the summer and colder in the winter, which is a bad thing for any living thing |
Journal entries from our work on well being

- Procrastination. I have procrastinated far too long and I have a short paper due tomorrow that I haven't started yet.

- Work. Good day at work had the right support and students were listening and behaving which was awesome and I was less exhausted than usual.

- Nervous and bored. Waiting for my interview....feeling nervous for the interview and bored cuz I've been waiting for an hour.

- Finished another interview. This interview went better than last week's, I guess I'll see next week if I get the job or not!

- Surprise presentation for CS 142. Went into class and found out everyone was presenting on their final projects today but I totally forgot about that and was ambushed.

- Hungry. Hungry and I don't want to eat junk food but we're at the aquarium and I have to

- Even more scones.. Vanilla chai this time. Delicious. Omg so many scones.

- Pouring rain. It's pouring outside and I have no umbrella because I lost mine.

- Tanned today!. It was sunny and really hot so I laid in the sun to tan. I was with Rene and Amy, I prefer to tan by myself so I left.

- Chipotle with Kyle. Kyle and I went to chipotle. We talked about the importance of family. It was a defining moment.

- Makeup brush broke!. The bristles of my Mac makeup brushes fell off of the handle and because I misplaced the receipt, I can't replace it. These things are expensive!

Which are Pos? which are Neg?
HW3 will use the restaurant data again.

File: 1 (17.html)
Review: 1
Atmosphere = 5
Do they accept reservations? = Yes
Food = 5
How many were seated at your table? = 2
Wine = "Red of some sort."
Overall = 5
Price/Value = 5
Service = 5
Text = "I think I may be the biggest cheerleader for the Tavolini anywhere, and here's why - ever since my wife and I started dating in high school, we've looked for somewhere to partake in that most-rare fine dining experience in Brantford, an otherwise terrible city to visit or live in. We first tried the Tavolini in the late 90's, and have returned several times, occasionally travelling halfway across the province to enjoy Anna's pasta, garlic bread and tartufo. We are treated like royalty every visit, even though we can never find a way to spend even a hundred dollars there - an amazing value for a full meal, with appetizers, drinks and desserts all around. The Tavolini is a truly special place, and if you have the misfortune to be passing through Brantford, you owe it to yourself to get off the highway, head downtown, lock the car up tight and enjoy some fine, fine food."

Was it a pleasant experience? = Yes
Would you return? = Yes

Review: 2
Atmosphere = 5
Do they accept reservations? = Yes
Food = 5
How many were seated at your table? = 4
Wine = "Valpolicella"
Overall = 5
Price/Value = 5
Service = 5
Text = "Best Italian food we ever had! Homemade fresh pasta and the veal dish my guest had was delicious! Very cozy, comfortable, homey atmosphere. The homemade tiramisu (the owner's grandmother's recipe) was out of this world! We will definitely be returning!"

Was it a pleasant experience? = Yes
Diagram of supervised classification

(a) Training
- Input
- Feature extractor
- Features
- Machine learning algorithm
- Label

(b) Prediction
- Input
- Feature extractor
- Features
- Classifier model
- Label

http://www.nltk.org/howto/classify.html
Our approach is TOOLs based
Setting up a classification experiment

- Any data set with at least two categories

- Where do we get the category LABELS?
  - For restaurant reviews, the reviewers provided them
  - For sarcasm tweets, we use the #sarcasm hashtag (and then remove it for learning)
  - For sarcasm forums, we collected annotations from 7 judges on Mechanical Turk
  - For Echo, well being, the users entered their happiness rating (1 to 9)

- People (practitioners) are always looking for ‘free data’

- In practice, **most of the time** we Turk
Mechanical Turk: A cottage industry

- Crowdsourcing is key to doing supervised classification and learning experiments.
- HIT = Human Intelligence Task
- A micro-task $0.25
- Everybody: industry and academics are doing it

<table>
<thead>
<tr>
<th>HIT Number</th>
<th>Description</th>
<th>Duration</th>
<th>Payment</th>
<th>Rating</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Answer a short survey about casino resorts - Tom Brennan</td>
<td>.40/3:00</td>
<td>(&gt;1000, &gt;95%)</td>
<td>Antanoni</td>
<td>5 minutes ago</td>
</tr>
<tr>
<td>2</td>
<td>Interested in opting-in to a special experiment? (Don't worry about opting-in to more than one experiment) - Sergey Schmidt</td>
<td>0.00/0:35</td>
<td>(&gt;1, &gt;90%)</td>
<td>Antanoni</td>
<td>10 minutes ago</td>
</tr>
<tr>
<td>3</td>
<td>Academic Survey (about 2-3 minutes in length) - Survey Researcher</td>
<td>$0.25/1:30</td>
<td>&gt;95%</td>
<td>self-HITWorthTurking</td>
<td>19 minutes ago</td>
</tr>
<tr>
<td>4</td>
<td>LMI Air Travel Price Survey - LMI Library</td>
<td>$1.00/2 min</td>
<td>(&gt;75%, &gt;100)</td>
<td>self-HITWorthTurking</td>
<td>43 minutes ago</td>
</tr>
<tr>
<td>5</td>
<td>Answer a short survey (less than ten minutes). - Ilyana Kuziemko</td>
<td>$1.00/4 min</td>
<td>(&gt;90%)</td>
<td>self-HITWorthTurking</td>
<td>Submitted an hour ago by sausxsx</td>
</tr>
<tr>
<td>6</td>
<td>Survey about hockey - many participants will qualify for additional bonus survey - Consumer Research Survey</td>
<td>$0.50 / &lt; 5 min</td>
<td>(&gt;95%)</td>
<td>self-HITWorthTurking</td>
<td>2 hours ago by deatthamrajgak</td>
</tr>
<tr>
<td>7</td>
<td>Word Generation Task and Company Evaluation Study - Melissa</td>
<td>$0.75/7 mins</td>
<td>(&gt;500, &gt;97%)</td>
<td>self-HITWorthTurking</td>
<td>Submitted 2 hours ago by badturky6</td>
</tr>
<tr>
<td>8</td>
<td>Daily Goals Survey - Stanford GSB Behavioral Lab</td>
<td>$0.50/4 min</td>
<td>&gt;95%</td>
<td>self-HITWorthTurking</td>
<td>Submitted 3 hours ago by self-HITWorthTurking</td>
</tr>
</tbody>
</table>
Mechanical Turk: NLDS a requester

Mechanical Turk is a marketplace for work. We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it’s convenient.

336,299 HITs available. View them now.

Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. Get Started.

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you’re satisfied with the results
One of our sarcasm HITs: easy interface

Below are 10 exchanges taken from an online discussion forum spanning a number of topics. The first utterance that is presented is the initial post of a forum thread, and the second is a response by a different user to that initial post. Some, all, or none of the responses to these quotes may be sarcastic.

We would like you to judge to your best ability whether or not any part of the response in each of these pairs appears to be sarcastic. If it is unclear, use your best judgement. We will be keeping track of worker reliability through hidden known answers and an automated algorithm.
How do we get the features?

- The things that we use to try to predict the labels?
- Use the tools we learned so far.
- Words, Stemmed words
- POS unigram and bigram counts
- Word endings “ful” “able”
- POS patterns “very ADJ” “not ADJ”
- Next week use Regexp, and sentiment words
Dev lets you test and refine without overfitting to your test
Training, Dev and Test

Overfitting ≈ seeing the exam before you take it
Text Classification Experiments

- Divide the corpus into three sets:
  - training set
  - test set
  - development (dev-test) set

1. LOOK AT YOUR DATA AND FORM HYPOTHESES ABOUT PATTERNS

2. Choose the features that will be used to classify the corpus.

3. Train the classifier on the training set.

4. Run it on the development set.

5. ANALYSE YOUR ERRORS: Refine the feature extractor from any errors produced on the development set.

6. REPEAT 1 THRU 4 UNTIL RUN OUT OF TIME OR IDEAS.

7. Run the improved classifier on the test set. CALCULATE YOUR FINAL RESULTS
Homework 3: Due next Monday

- Worth 10 points
- Practice everything you know. Unigrams, Bigrams, POS
- Do an initial text classification experiment
  - We set up training, development and test sets for the restaurant reviews.
  - You figure out what features you can extract
  - You test on development and try to make it better.
  - We test it on the test set.
  - **Competition**: see who can get the best accuracy on the test set.
- Then the following week we add more features and try again on this data set and a new one
What the representation looks like

Vectors of features, the label

```python
>>> train = [
    ...     (dict(a=1,b=1,c=1), 'y'),
    ...     (dict(a=1,b=1,c=1), 'x'),
    ...     (dict(a=1,b=1,c=0), 'y'),
    ...     (dict(a=0,b=1,c=1), 'x'),
    ...     (dict(a=0,b=1,c=1), 'y'),
    ...     (dict(a=0,b=0,c=1), 'y'),
    ...     (dict(a=0,b=1,c=0), 'x'),
    ...     (dict(a=0,b=0,c=0), 'x'),
    ...     (dict(a=0,b=1,c=1), 'y'),
    ...]

>>> test = [
    ...     (dict(a=1,b=0,c=1)), # unseen
    ...     (dict(a=1,b=0,c=0)), # unseen
    ...     (dict(a=0,b=1,c=1)), # seen 3 times, labels=y,y,x
    ...     (dict(a=0,b=1,c=0)), # seen 1 time, label=x
    ...
    ...]
```
There are lots of different classifiers

- They are all **different ways to learn a function**
  - F(feature vector) => Label

- F can be linear, or more complex.
  - Naïve Bayes
  - Rule Induction
  - Linear Regression
  - Tree regression
  - Classification and Regression Trees

- Let me show you some examples.
Naïve Bayes

- Stack Overflow explanation of NB
Also can predict scalars: Linear Regression
Personality Classification:

Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text

François Mairette
Department of Computer Science, University of Sheffield,
211 Portobello Street, Sheffield S1 4DP, United Kingdom
F.MAIRESSE@SHEFFIELD.AC.UK

Marilyn A. Walker
Department of Computer Science, University of Sheffield
211 Portobello Street, Sheffield S1 4DP, United Kingdom
M.A.WALKER@SHEFFIELD.AC.UK

Matthias R. Mehl
Department of Psychology, University of Arizona,
1503 E University Blvd. Building 68, Tucson, AZ 85721, USA
MEHL@EMAIL.ARIZONA.EDU

Roger K. Moore
Department of Computer Science, University of Sheffield,
211 Portobello Street, Sheffield S1 4DP, United Kingdom
R.K.MOORE@DCS.SHEF.AC.UK
<table>
<thead>
<tr>
<th>Introvert</th>
<th>Extravert</th>
</tr>
</thead>
<tbody>
<tr>
<td>- I don't know man, it is fine I was just saying I don't know.</td>
<td>- Oh, this has been happening to me a lot lately. Like my phone will ring.</td>
</tr>
<tr>
<td>- I was just giving you a hard time, so.</td>
<td>It won't say who it is. It just says call. And I answer and nobody will say anything. So I don't know who it is.</td>
</tr>
<tr>
<td>- I don't know.</td>
<td>- Okay. I don't really want any but a little salad.</td>
</tr>
<tr>
<td>- I will go check my e-mail.</td>
<td></td>
</tr>
<tr>
<td>- I said I will try to check my e-mail, ok.</td>
<td></td>
</tr>
</tbody>
</table>
Figure 4: M5' regression tree for observed conscientiousness, computed using the EAR corpus. The target output ranges from 1 to 7, where 7 means strongly conscientious (Comm. words is the ratio of words related to communication).
Does a decision tree define a linear function?

What are the splits at the nodes doing?
Disagreement Decision Tree: How to read it

- LIWC:Total second person $\geq 0.7$ and LIWC:Sentences ending with "?" $\geq 3.8$
  - True: Disagree (163.0/20.0)
  - False:
    - LIWC:Total second person $\geq 0.7$ and LIWC:Negations $\geq 1.2$
      - True: LIWC:Metaphysical issues $\geq 1.7$ and LIWC:Negations $\geq 2.7$
        - True: LIWC:Sentences ending with "?" $\geq 16.7$
          - True: Disagree (47.0/14.0)
          - False: Agree (435.0/106.0)
        - False: Disagree (39.0/8.0)
      - False: Disagree (136.0/39.0)
    - False: Disagree (39.0/8.0)
Learning Decision Rules for Personality

- Rules and Trees easy to understand
- Different learners can give very different results

<table>
<thead>
<tr>
<th>#</th>
<th>Ordered rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((\text{School} \geq 1.47) \land (\text{Motion} \geq 1.71) \Rightarrow \text{NOT OPEN})</td>
</tr>
<tr>
<td>2</td>
<td>((\text{Occup} \geq 2.49) \land (\text{Sixlitr} \leq 13.11) \land (\text{School} \geq 1.9) \land (I \geq 10.5) \Rightarrow \text{NOT OPEN})</td>
</tr>
<tr>
<td>3</td>
<td>((\text{Fam} \geq 600.335106) \land (\text{Friends} \geq 0.67) \Rightarrow \text{NOT OPEN})</td>
</tr>
<tr>
<td>4</td>
<td>((\text{Nnet} \leq 3.502543) \land (\text{Number} \geq 1.13) \Rightarrow \text{NOT OPEN})</td>
</tr>
<tr>
<td>5</td>
<td>((\text{School} \geq 0.98) \land (\text{You} \leq 0) \land (\text{AllPct} \leq 13.4) \Rightarrow \text{NOT OPEN})</td>
</tr>
<tr>
<td>6</td>
<td>Any other feature values \Rightarrow \text{OPEN}</td>
</tr>
</tbody>
</table>

Table 16: JRip rule set for binary classification of openness to experience, based on the essays corpus.

and 5). As expected, the avoidance of longer words is also indicative of a lack of creativity/conventionality (Rules 4 and 5), as well as the use of high-familiarity words and references to friends (Rule 3).
Choosing the right features

- Unlike just looking at your data and trying to form hypotheses about patterns, classifiers come with tools that help you figure out what features are helping and which are not.

- Use too few, too general, and the data will be **underfitted**.
  - The classifier is too vague and makes too many mistakes.

- Use too many, too specific, and the data will be **overfitted**.
  - The classifier is too specific and will not generalize to new examples.
Classification: Using Naïve Bayes
(other classifiers similar)
Classifiers label tokens with category labels (or *class labels*). Typically, labels are represented with strings (such as "health" or "sports"). In NLTK, classifiers are defined using classes that implement the `ClassifierI` interface:

```python
>>> import nltk
>>> nltk.usage(nltk.classify.ClassifierI)
ClassifierI supports the following operations:
    - self.classify(featureset)
    - self.classify_many(featuresets)
    - self.labels()
    - self.prob_classify(featureset)
    - self.prob_classify_many(featuresets)
```

NLTK defines several classifier classes:

- `ConditionalExponentialClassifier`
- `DecisionTreeClassifier`
- `MaxentClassifier`
- `NaiveBayesClassifier`
- `WekaClassifier`

Classifiers are typically created by training them on a training corpus.
Text Classification Experiments

- Divide the corpus into three sets:
  - training set
  - test set
  - development (dev-test) set

1. **LOOK AT YOUR DATA AND FORM HYPOTHESES ABOUT PATTERNS**

2. Choose the features that will be used to classify the corpus.

3. Train the classifier on the training set.

4. Run it on the development set.

5. **ANALYSE YOUR ERRORS**: Refine the feature extractor from any errors produced on the development set.

6. **REPEAT 1 THRU 4 UNTIL RUN OUT OF TIME OR IDEAS**.

7. Run the improved classifier on the test set. **CALCULATE YOUR FINAL RESULTS**
What gender is a name?

- Men’s and women’s names tend to pattern differently.
- If you didn’t know could you predict from name features?

1. Sophia
2. Emma
3. Olivia
4. Isabella
5. Mia
6. Ava
7. Lily
8. Zoe
9. Emily
10. Chloe
11. Layla
12. Madison
13. Benjamin
14. Michael
15. Caleb
16. Ryan
17. Alexander
18. Elijah
19. James
20. William
21. Oliver
22. Connor
23. Matthew
13. Madelyn
14. Abigail
15. Aubrey
16. Charlotte
17. Amelia
18. Ella
19. Kaylee
20. Avery
21. Aaliyah
22. Hailey
23. Hannah
Feature Extraction: NLTK Dictionary

- Gender example from book: Sec 6.1
- Last letter of name is a good feature
- Make a ‘dict’ feature-name: value

```python
>>> def gender_features(word):
    ...     return {'last_letter': word[-1]}
>>> gender_features('Shrek')
{'last_letter': 'k'}
```

The returned dictionary, known as a feature set, maps from features' names to their values. Feature names are case-sensitive strings that typically provide a short human-readable description of the feature. Feature values are values with simple types, such as booleans, numbers, and strings.
Once we’ve done this the ‘classifier’ is ‘trained model’ for predicting the gender of a name.
Then we can test our ‘trained model’ on new names we haven’t seen before
And we can test on a whole batch.

What is Accuracy?

Let’s say we have 100 in our test, evenly split. The table shows the counts of actual and predicted gender.

<table>
<thead>
<tr>
<th>Actual/Predicted</th>
<th>Female</th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>Male</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

What is the Accuracy?

```
>> print(nltk.classify.accuracy(classifier, test_set))
.758
```
And we can test on a whole batch

```
>> print(nltk.classify.accuracy(classifier, test_set))
0.758
```

What is Accuracy?
Let’s say we have 100 in our test, evenly split

<table>
<thead>
<tr>
<th>Actual/Predicted</th>
<th>Female</th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>Male</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

This is called a confusion matrix.
What is the Accuracy?
Is accuracy what we always care about?

- What if it was a problem like diagnosing cancer?

<table>
<thead>
<tr>
<th>Actual/Predicted</th>
<th>Has Cancer</th>
<th>Doesn’t have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has Cancer</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>Doesn’t have</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

- Are both kinds of errors the same?
Is accuracy what we always care about?

- What if it was a problem like diagnosing cancer?

<table>
<thead>
<tr>
<th>Actual/Predicted</th>
<th>Has Cancer</th>
<th>Doesn’t have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has Cancer</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>Doesn’t have</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

- For some problems a false positive is okay but a false negative may not be.

- Other measures besides accuracy we will use:
  - **Precision**: for the category you care about, if you said the item was that category, were you right?
  - **Recall**: for the category that you care about did you find all the ones that were there?
More useful measures

- Precision = $\frac{TP}{TP + FP}$
- Recall = $\frac{TP}{TP + FN}$
- F-Measure = $\frac{2 \times Precision \times Recall}{Precision + Recall}$
Informative features: examine the model

- Usually want to look at more than just the top five features
- 38 times more likely to see “a” as the last letter of a female name
- 31 times more likely to see “k” as the last letter of a male name
- Classifiers often work better with fewer features
Its Creative!

- Figuring out how to represent a problem and what features to use is a big aspect of creativity with NLP problems.

- How to encode your intuition is the root of the problem (into a vector!!)

- How to test your intuitions

- How to figure out if your intuitions are wrong, or whether it’s the learner or the way you’ve encoded it.

- Tools you can use to figure it out

- **Looking at your data and analyzing errors on the dev set**