Lecture 21:
Unlabeled data for NLP

Intro to NLP, CS585, Fall 2014
http://people.cs.umass.edu/~brenocon/inlp2014/
Brendan O’Connor
• Project scheduling
• Labeling
• What to do when we only have a little bit of labeled data? (Like in the final project!)
  • Get more labels
  • Different forms of supervision
    • Tag dictionaries: type-level supervision
    • More sophisticated features
  • Exploit unlabeled data
    • Semi-supervised learning
    • Active learning: intelligently choose which unlabeled data to annotate
Unlabeled data

- Labeled data: human element is costly
  - PTB or ImageNet: the largest labeled datasets and very successful -- but very expensive!
    - PTB = 1M tokens
    - ImageNet = 1M images
- Small efforts and new problems: typically thousands of tokens
- But we have huge quantities of unlabeled, raw text. Can we use them somehow?
45k tokens
(our NER dataset)
45k tokens (our NER dataset)

IM tokens (WSJ PTB)
Semi-supervised learning

• Formally: given
  • (1) small labeled dataset of (x,y) pairs,
  • (2) large unlabeled dataset of (x, _) pairs,
  • ... learn a better f(x)->y function than from just labeled data alone.

• Two major approaches
  • 1. Learn an unsupervised model on the x’s. Use its clusters/vectors as features for labeled training.
  • 2. Learn a single model on both labeled and unlabeled data together
Unsupervised NLP

• Can we learn lexical or grammatical structures from unlabeled text?
  • Maybe lexical/structural information is a latent variable ... like alignments in IBM Model I
  • (Different use: exploratory data analysis)
• Intuition for lexical semantics: the distributional hypothesis.
  • You shall know a word by the company it keeps (Firth, J. R. 1957:11)
• Very useful technique: learn word clusters (or other word representations) on unlabeled data, then use as features in a supervised system.
Distributional example:
What types of words can go into these positions?

- the ____
- that ____
- of ____
- by ____

- he __
- she __
- Mary __
- John __

- red ___
- green ___

- happy ___
- angry ___
- sad ___

- __ it
- __ him
- __ her

- __ lol
- __ haha

Distributional semantics is based on the idea that:
Words with similar context statistics have similar meaning.

Assemble sets of words with similar context frequencies.

Many ways to capture this... including HMMs.
Brown HMM word clustering

- HMM for the unlabeled dataset
  - With a one-class-per-word restriction!
    - (Remember: real-world POS data kinda has this property)
  - Thus each HMM class is described by a hard clustering of words (a set of words)
- Heuristically search for word clusters that maximize likelihood

Notation:
c is a clustering of wordtypes. $c(w)$ is w’s cluster ID.

$$c^* = \arg \max_{c \in C} \prod_i p_{\text{MLE}}(c(w_i) \mid c(w_{i-1})) \times p_{\text{MLE}}(w_i \mid c(w_i))$$
Hierarchical clustering

• One form of Brown clustering is also hierarchical, through agglomerative clustering: iteratively merge clusters, and track the merge history
  • Initialize: Greedily assign words to K clusters
  • Iterate: Merge the two clusters that causes the least-worst hit to likelihood

• (There are many other approaches to this type of HMM; see http://statmt.blogspot.com/2014/07/understanding-mkcls.html)
Brown Algorithm

- Words merged according to contextual similarity
- Clusters are equivalent to bit-string prefixes
- Prefix length determines the granularity of the clustering

[Slide credit: Terry Koo]
Brown Algorithm

- Words merged according to contextual similarity
- Clusters are equivalent to bit-string prefixes
- Prefix length determines the granularity of the clustering
Brown Algorithm

• Words merged according to contextual similarity

• Clusters are equivalent to bit-string prefixes

• Prefix length determines the granularity of the clustering

[Slide credit: Terry Koo]
Hier. clusters as POS features

- 1000 leaves, cluster prefixes as features for Twitter POS

Using the Liang 2005 version of Brown clustering: https://github.com/percyliang/brown-cluster

Highest Weighted Clusters

<table>
<thead>
<tr>
<th>Cluster prefix</th>
<th>Tag</th>
<th>Types</th>
<th>Most common word in each cluster with prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>11101010*</td>
<td>!</td>
<td>8160</td>
<td>lol lmao haha yes yea oh omg aww ah btw wow thanks sorry congrats welcome yay ha hey goodnight hi dear please huh wtf exactly idk bless whatever well ok</td>
</tr>
<tr>
<td>11110*</td>
<td>A</td>
<td>6510</td>
<td>young sexy hot slow dark low interesting easy important safe perfect special different random short quick bad crazy serious stupid weird lucky sad</td>
</tr>
<tr>
<td>1101*</td>
<td>D</td>
<td>378</td>
<td>the da my your ur our their his</td>
</tr>
<tr>
<td>01*</td>
<td>V</td>
<td>29267</td>
<td>do did kno know care mean hurts hurt say realize believe worry understand forget agree remember love miss hate think thought knew hope wish guess bet have</td>
</tr>
<tr>
<td>11101*</td>
<td>O</td>
<td>899</td>
<td>you yall u it mine everything nothing something anyone someone everyone nobody</td>
</tr>
<tr>
<td>100110*</td>
<td>&amp;</td>
<td>103</td>
<td>or n &amp; and</td>
</tr>
</tbody>
</table>

http://www.ark.cs.cmu.edu/TweetNLP/cluster_viewer.html
Other examples

- Dependency parsing (Koo et al. 2008)

<table>
<thead>
<tr>
<th>Training Sentences</th>
<th>Baseline</th>
<th>Cluster-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>82.0</td>
<td>85.3 (+3.3)</td>
</tr>
<tr>
<td>2000</td>
<td>85.0</td>
<td>87.5 (+2.5)</td>
</tr>
<tr>
<td>4000</td>
<td>87.9</td>
<td>89.7 (+1.8)</td>
</tr>
<tr>
<td>8000</td>
<td>89.7</td>
<td>91.4 (+1.7)</td>
</tr>
<tr>
<td>16000</td>
<td>91.1</td>
<td>92.2 (+1.1)</td>
</tr>
<tr>
<td>32000</td>
<td>92.1</td>
<td>93.2 (+1.1)</td>
</tr>
<tr>
<td>39832</td>
<td>92.4</td>
<td>93.3 (+0.9)</td>
</tr>
</tbody>
</table>

- NER (Miller et al. 2004)

This is a learning curve analysis: performance as a function of training set size
Brown clusters as features

- Have been seen useful for
  - POS
  - NER
  - Dependency parsing
  - (others?)

- More generally: use automatically learned
  word representations. Next week: vector-valued reprs.
- I think word reprs are the most established use of
  unlabeled data for NLP systems

See also: http://metaoptimize.com/projects/wordreprs/
Semi-supervised learning

- Formally: given
  - (1) small labeled dataset of \((x, y)\) pairs,
  - (2) large unlabeled dataset of \((x, _)\) pairs,
  - ... learn a better \(f(x) \rightarrow y\) function than from just labeled data alone.

- Two major approaches
  1. Learn an unsupervised model on the x’s. Use its clusters/vectors as features for labeled training.
  2. Learn a single model on both labeled and unlabeled data together
EM for semi-sup learning

• we have
  • (1) small labeled dataset of (x,y) pairs,
  • (2) large unlabeled dataset of (x, _) pairs,

• Treat missing labels as latent variables. Learn with EM!
  • Init: train model on labeled data
  • E-step: soft predictions on unlabeled
  • M-step: maximize labeled loglik, PLUS weighted loglik according to our new soft predictions. So the entire unlabeled dataset is part of the training set

• Issues:
  • Have to re-weight the M-step (what if unlabeled data is 1 million times bigger?)
  • Can go off the rails
Self-training

• Same setup, but only add in a small number of highly-confident examples
  • Label all unlabeled x’s. Choose the top-10 most confident (and/or higher than 99% confidence...).
  • Add those 10 to the labeled dataset
  • Re-train and iterate
• Many examples of this being useful -- may have to limit the number of iterations and/or play with thresholds
  • E.g. best parsers use self-training
Active learning

- You want to label more data. Use your current classifier to help choose the most useful examples to annotate.
- **Uncertainty sampling**: Choose the example where the model is most uncertain. (If binary: closest to 50% predicted prob. If multiclass: highest entropy)

(a) a 2D toy data set

- My take: some people in industry swear by AL, but I haven’t seen many research papers showing dramatic gains from it. Not sure why the difference. See review by [http://burrsettles.com/](http://burrsettles.com/)
Active learning

- You want to label more data. Use your current classifier to help choose the most useful examples to annotate.
- **Uncertainty sampling**: Choose the example where the model is most uncertain. (If binary: closest to 50% predicted prob. If multiclass: highest entropy)

![A 2D toy data set](image1)

(a) a 2D toy data set

![Random sampling](image2)

(b) random sampling

- My take: some people in industry swear by AL, but I haven’t seen many research papers showing dramatic gains from it. Not sure why the difference. See review by [http://burrsettles.com/](http://burrsettles.com/)
Active learning

- You want to label more data. Use your current classifier to help choose the most useful examples to annotate.
- **Uncertainty sampling**: Choose the example where the model is most uncertain. (If binary: closest to 50% predicted prob. If multiclass: highest entropy)

(a) a 2D toy data set  
(b) random sampling  
(c) uncertainty sampling

- My take: some people in industry swear by AL, but I haven’t seen many research papers showing dramatic gains from it. Not sure why the difference. See review by [http://burrsettles.com/](http://burrsettles.com/)