CSCI 5832
Natural Language Processing

Lecture 15
Jim Martin

Today 3/8

- **Review Sequence Labeling/Chunking**
- **More on Classifiers**
- **Break**
- **Project discussions**

Statistical Sequence Labeling

- As with POS tagging, we can use rules to do partial parsing or we can train systems to do it for us. To do that we need training data and the right kind of encoding.
  - Training data
    - Hand tag a bunch of data (as with POS tagging)
    - Or even better, extract partial parse bracketing information from a treebank.

Encoding

- With the right encoding you can turn the labeled bracketing task into a tagging task. And then proceed exactly as we did with POS Tagging.
- We'll use whats called IOB labeling to do this.
  - I -> Inside
  - O -> Outside
  - B -> Begins

IOB encoding

The morning flight from Denver has arrived.
B_NP I_NP I_NP O B_NP O O

The morning flight from Denver has arrived
B_NP I_NP I_NP B_PP B_NP B_VP I_VP

- The first example shows the encoding for just base-NPs. There are 3 tags in this scheme.
- The second shows full coverage. In this scheme there are 2*N+1 tags. Where N is the number of constituents in your set.

Methods

- HMMs

$$\begin{aligned}\hat{T} &= \operatorname{argmax}_T P(T|W) \\ &= \operatorname{argmax}_T P(W|T)P(T) \\ &= \operatorname{argmax}_T \prod_i P(word_i|tag_i) \prod_i P(tag_i|tag_{i-1})\end{aligned}$$

- Sequence Classification
  - Using any kind of standard ML-based classifier.

Evaluation

- Suppose you employ this scheme. What's the best way to measure performance.
- Probably not the per-tag accuracy we used for POS tagging.
  - Why?

•It's not measuring what we care about
•We need a metric that looks at the chunks not the tags

Precision/Recall/F

- Precision:
  - The fraction of chunks the system returned that were right
    - "Right" means the boundaries and the label are correct given some labeled test set.
- Recall:
  - The fraction of the chunks that system got from those that it should have gotten.
- F: Harmonic mean of those two numbers.

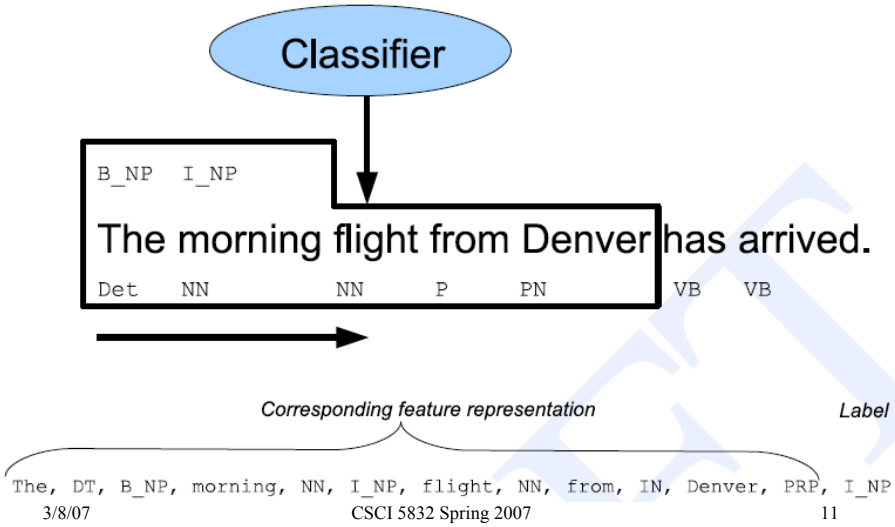
Supervised Classification

- Training a system to take an object represented as a set of features and apply a label to that object.
- Methods typically include
  - Naïve Bayes
  - Decision Trees
  - Maximum Entropy (logistic regression)
  - Support Vector Machines
  - …

Supervised Classification

- Applying this to tagging…
  - The object to be tagged is a word in the sequence
  - The features are features of the word, features of its neighbors, and features derived from the entire sentence.
  - Sequential tagging means sweeping the classifier across the input assigning tags to words as you proceed.

Statistical Sequence Labeling



Typical Features

- **Typical setup involves**
  - **A small sliding window around the object being tagged**
  - **Features extracted from the window**
    - **Current word token**
    - **Previous/next N word tokens**
    - **Current word POS**
    - **Previous/next POS**
    - **Previous N chunk labels**
      - **????**

Performance

- With a decent ML classifier
  - SVMs
  - Maxent
  - Even decision trees
- You can get decent performance with this arrangement.
- Good CONLL 2000 scores had F-measures in the mid-90s.

Problem

- You're making a long series of local judgments, without attending to the overall goodness of the final sequence of tags. You're just hoping that local conditions will yield global optima.
- Note that HMMs didn't have this problem since the language model worried about the overall goodness of the tag sequence.

Answer

- Graft a language model onto the sequential classification scheme.
  - Instead of having the classifier emit one label as an answer, get it to emit an N-best list for each judgment.
  - Run viterbi over the N-best lists for the sequence to get the best overall sequence.

MEMMs

- Maximum entropy Markov models are the current standard way of doing this.
  - Although people do the same thing in an ad hoc way with SVMs.
- MEMMs combine two techniques
  - Maximum entropy (logistic) classifiers for the individual labeling
  - Markov models for the sequence model.

Models

- HMMs and graphical models are often referred to as generative models since they're based on using Bayes…
  - So to get P(c|x) we use P(x|c)P(c)
- Alternatively we could use what are called discriminative models; models that get P(c|x) directly without the Bayesian inversion

MaxEnt

- Multinomial logistic regression
- Along with SVMs, Maxent is the typical technique used in NLP these days when a classifier is required.
  - Provides a probability distribution over the classes of interest
  - Admits a wide variety of features
  - Permits the hand-creation of complex features
  - Training time isn't bad

MaxEnt

$$p(c|x) = \frac{1}{Z} \exp \sum_i w_i f_i$$

MaxEnt

$$p(c|x) = \frac{\exp\left(\sum_{i=0}^{N} w_{ci} f_i\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)}$$

Hard Classification

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|x)$$

- If we really want an answer…
- But typically we want a distribution over the answers.

MaxEnt Features

- They're a little different from the typical supervised ML approach
  - Limited to binary values
    - Think of a feature as being on or off rather than as a feature with a value
  - Feature values are relative to an object/class pair rather than being a function of the object alone.
  - Typically have lots and lots of features (100,000s of features isn't unusual.)

Features

$$f_3(c,x) = \begin{cases} 1 & \text{if suffix}(word_i) = \text{“ing”} \ \& \ c = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$f_4(c,x) = \begin{cases} 1 & \text{if is\_lower\_case}(word_i) \ \& \ c = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

Features

$$f_3(c,x) = \begin{cases} 1 & \text{if suffix}(word_i) = \text{“ing”} \ \& \ c = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$f_4(c,x) = \begin{cases} 1 & \text{if is\_lower\_case}(word_i) \ \& \ c = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

- **Key point. You can't squeeze features like these into an HMM.**

Mega Features

$$f_{125}(c,x) = \begin{cases} 1 & \text{if } word_{i-1} = \texttt{<s>} \ \& \ \text{isupperfirst}(word_i) \ \& \ c = \text{NNP} \\ 0 & \text{otherwise} \end{cases}$$

- **These have to be hand-crafted.**
- **With the right kind of kernel they can be exploited implicitly with SVMs. At the cost of a increase in training time.**

Back to Sequences

$$\begin{aligned} \hat{T} &= \underset{T}{\text{argmax}} P(T|W) \\ &= \underset{T}{\text{argmax}} P(W|T)P(T) \\ &= \underset{T}{\text{argmax}} \prod_i P(word_i|tag_i) \prod_i P(tag_i|tag_{i-1}) \end{aligned}$$

- **HMMs**

$$\begin{aligned} \hat{T} &= \underset{T}{\text{argmax}} P(T|W) \\ &= \underset{T}{\text{argmax}} \prod_i P(tag_i|word_i, tag_{i-1}) \end{aligned}$$

- **MEMMs**

And whatever other features you choose to use!

Back to Viterbi

$$v_t(j) = \max_{1 \le i \le N-1} v_{t-1}(i)\, P(s_j|s_i, o_t); \quad 1 < j < N, 1 < t < T$$

- The value for a cell is found by examining all the cells in the previous column and multiplying by the posterior for the current column (which incorporates the transition as a factor, along with any other features you like).

Break

- Next Quiz
  - Chapters 11 and 12 and parts of 6 and 13. I'll post the exact readings
  - Next Thursday 3/15
- Change in schedule
  - Adding 2 lectures on NLP for bioinformatics in the week before break (3/20 and 22).
  - I'll rearrange things to make it fit.