Notes on “Nymble: a High-Performance Learning Name-finder”

Brendan Linn

January 29, 2009
Overview

- Supervised learning
  - Named entities in the training set are tagged with SGML
- Assume a HMM with states for each name-class (organization names, person names, location names, times, dates, percentages, money amounts, etc.)
- Use training set to estimate probabilities of
  - Current name-class, given previous name-class and previous word
  - Initial word in name-class, given current and previous name-classes
  - Non-initial word in name-class, given previous word
- Use Viterbi algorithm to construct a sequence of name-classes most likely to correspond to input sequence
HMM Structure

person -¿ end; org -¿ end; etc -¿ end; notname -¿ end;
person -¿ person; org -¿ org; etc -¿ etc; notname -¿ notname; ;
Background

- Want to find a sequence of name-classes $NC$ that maximizes $Pr[NC \mid W] = Pr[NC, W] / Pr[W]$, where $W$ is the observed sequence of words.

- But authors assume $Pr[w \in W] = 1$ all the time, so that we just want to maximize $Pr[NC, W]$.
  - Is this a good assumption?

- A word is an ordered pair $\langle w, f \rangle$, where $w$ is a string and $f$ is a simple feature computed from the string. E.g., $w = 23,000.00$ might have $f = \text{containsDigitAndComma}$.
  - $f$ is used for heuristics, e.g. $f = \text{allCaps}$ might indicate an organization name.
Generate current name-class and initial word:

$$\Pr [NC | NC_{-1}, w_{-1}] \times \Pr [\langle w, f \rangle | NC, NC_{-1}]$$

- Why $w_{-1}$ instead of $\langle w, f \rangle_{-1}$? Just intuition.

Generate non-initial word in current name-class:

$$\Pr [\langle w, f \rangle | \langle w, f \rangle_{-1}, NC]$$

Generate final word in current name-class:

$$\Pr [\text{end} | \langle w, f \rangle_{-1}, NC],$$

where end is a fake end-of-name-class word.
Training Problems

- What if a word $\langle w, f \rangle$ is observed but $\langle w, f \rangle \not\in V$, where $V$ is the vocabulary of the training set?

- Technique: use one half of the training set to build $V$, then use the other half of to estimate $\Pr[\text{unknown} | \langle w, f \rangle_{-1}]$, $\Pr[\langle w, f \rangle | \text{unknown}_{-1}]$, and $\Pr[\text{unknown} | \text{unknown}_{-1}]$ for each name-class. Then switch the halves and repeat.
Note that each name-class is supposed to be a complete directed graph on $|V|$ vertices, i.e. $|V|^2$ transitions per name-class.

What if some of those transitions aren’t observed during training?

Technique: back off to progressively weaker estimates. For example, if the bigram probability $Pr[\langle w, f \rangle | \langle w, f \rangle_{-1}, NC]$ is not observed during training, back off to the simpler unigram probability $Pr[\langle w, f \rangle | NC]$.

Actually, we adjust the weights between the best model and the back-off model.
Results

- Use Viterbi algorithm to reconstruct the sequence of name-class transitions with maximum probability in time linear in the number of tokens in the sentence.

- Measuring performance:

  \[ P = \frac{\text{correct responses}}{\text{total responses}} \]

  \[ R = \frac{\text{correct responses}}{\text{responses in key}} \]

  \[ F \approx \frac{2RP}{R + P} \]

- Authors report \( F \) in low 90s on English and Spanish.