Assignment 3
CS562
Prof. Kevin Knight
Due at the beginning of class Thursday, October 5, 2006

Part of Speech Tagging

In this assignment, you will write a program to automatically tag sequences of words with their parts of speech (noun, verb, determiner, etc.).

Training data is provided in: `/auto/home-scf-22/csci562/asst3/train-data`

NOTE: w = word
t = tag

Part 1. (5 points)

Build a P(w…w | t…t) channel model in Carmel WFST format. This model should be context independent, i.e., P(w…w | t…t) = P(w1 | t1) * P(w2 | t2) * … * P(wn | tn). Draw a sketch of the model and turn it in on paper. Email the WFST to `jonmay@isi.edu`.

Built two P(t…t) tag language models – unigram and bigram – in Carmel WFSA format. Draw sketches of the models and turn them in on paper. Email the WFSAs to `jonmay@isi.edu`.

Use Carmel to compute optimal tag sequences using (1) the unigram model + channel model, and (2) the bigram model + channel model. Test the two tagging systems on the sentences in `test-data-1.sent` and turn in the results. All of the words in that test set also appear in the training data.

> This is a test .
>
> The fly knows how to fly .
>
> The company has agreed to release its tax returns since 1985 , and those of its affiliates and partnerships .

- `% carmel -brIEQk 5 unigram.wfsa tag-to-word.wfst test-data-1.sent`
- `% carmel -brIEQk 5 bigram.wfsa tag-to-word.wfst test-data-1.sent`

Part 2. (10 points)

Test the two taggers on the larger 1000-word test set called `test-data-2.sent`.

- Show and compare results on the first five sentences

```
% carmel -brIEQk 1 unigram.wfsa tag-to-word.wfst test-data-2.sent | head -5
% carmel -brIEQk 1 bigram.wfsa tag-to-word.wfst test-data-2.sent | head -5
```

- Report the overall word-tagging accuracies for each of the two systems on all 1000 words. Correct tags are given in the file `test-data-2.correct`. A tagger that incorrectly tags 200 of the 1000 words would receive an accuracy of 80%.

NOTE: In this part, you will need to deal with previously-unseen words -- words in the 1000-word test set with zero occurrences in the training data. A simple way to do this is to add (to your channel model) small-weighted transitions that assign all possible tags to all test set words. There are variations on this theme, e.g., only allow certain tags to generate unknown words, use the spelling of a particular word to reduce the "possible tags" to a smaller group, train probabilities relating to unknown words, etc.

Part 3. (35 points)

Implement a Viterbi decoder for the bigram model and channel model. Tag the test sentences using this decoder instead of Carmel. By email to `jonmay@isi.edu`, turn in: (1) your code, (2) the results of tagging the first five sentences of test-data-2.sent, and (3) a trace of the Viterbi decoder running for one sentence.