

NLP Assignment 1:

David Samuelson, 208788851

Handling unknown words in HMM1:

Given unknown word, the e score I gave to the word was $\frac{\#tags}{\#words}$.

I found this score as a pretty good one. For example, $\frac{1}{\#words}$ is worse when measured by token accuracy.

Pruning Strategy in Viterbi HMM:

This pruning strategy was advised to me by on of the classmates.

The idea is – the tags that I need to choose from to tag the current word, are not the whole tag set. Instead, I iterate on the tags I have seen associated to this word in the training corpus.

In addition – the previous and previous previous tags are chosen from the tags I have checked before. May be described as "Sliding window" of tags.

Test Scores:

- NER Accuracies:
 - HMM
 - Greedy
 - Per token accuracy: 0.993349877855
 - Per span accuracy: 0.929737484974
 - Viterbi
 - Per token accuracy: 0.991740664624
 - Per span accuracy: 0.940265229361
 - MEMM
 - Greedy
 - Per token accuracy: 0.993330489744
 - Per span accuracy: 0.933207956881
 - Viterbi
 - Per token accuracy: 0.993097832409
 - Per span accuracy: 0.93720190779
- POS-Tagging Accuracies:
 - HMM
 - Greedy
 - Per token accuracy: 0.998928135205
 - Viterbi
 - Per token accuracy: 0.998928135205
 - MEMM
 - Greedy
 - Per token accuracy: 0.998753645587
 - Viterbi
 - Per token accuracy: 0.998629010145

Difference between HMM and MaxEnt Taggers:

MaxEnt taggers are more flexible – I can add features or remove pretty easily, and support wide context.

HMM taggers, on the other side, are more limited. Once the score function defined – the context is "closed".

I think, but not sure, that HMM taggers are more resistant to changes, while small change in the MaxEnt tagger features causes significant change in the results. I saw this on a single example, when providing unnecessary features – the result significantly changed, while when changing score function of the HMM tagger – changes were minor.

Difference between datasets:

NER dataset has much less tags, therefore the per-token accuracy was very high.

Change in HMM Tagger to improve accuracy:

Change the scoring function, find the best parameters.

Another thing that can be changed is the score function – add more context (next words).

Change in MEMM Tagger to improve accuracy:

Currently, I am treating each word as non-rare word.

Possible thing to do is to give features to the word both as rare and non-rare word, and choose the best tag according to the two predictions by those features.

Another idea, like before – combine ALL features in one sample to predict.

Why span scores lower than accuracy scores:

Obviously, spans are a combination of multiple tokens together. One bad tag could ruin a whole span, while barely damaging the per token accuracy. There are less spans than tags, therefore span accuracy is lower.