Ameen Ali – 318429792

Tal Klein – 312592520

Intro to Natural Language Processing

Assignment 1 - Writeup

1. Describe how you handled unknown words in hmm1.

We mapped low frequency words which never seen in the training data after a certain threshold. In our case, we chose the threshold to be the last 10% of the training data and marked new words as UNK with the tag attached.

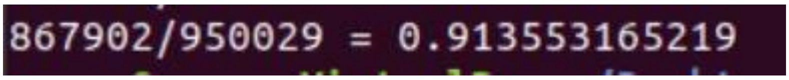
2. Describe your pruning strategy in the viterbi hmm.

If a word was seen in the training set – only it's tags are checked.
If a word was not seen in the training set – we tried to find the right signature of the word and matched the likely to be true tags. If we did not find a signature, we used the entire tag set. In this way, we have reduced the complexity of the algorithm.

3. Report your test scores when running the each tagger (hmm-greedy, hmm-viterbi, maxent-greedy, memm-viterbi) on each dataset. For the NER dataset, report token accuracy, as well as span precision, recall and F1.

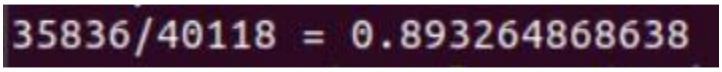
POS

HMM Greedy – training data (ass1-tagger-train)



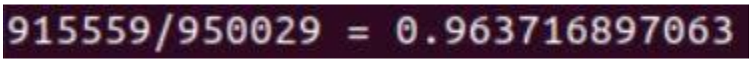
```
867902/950029 = 0.913553165219
```

HMM Greedy – test (ass1-tagger-test)



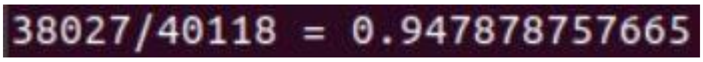
```
35836/40118 = 0.893264868638
```

HMM Viterbi – training data (ass1-tagger-train)



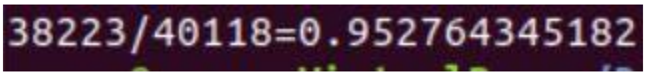
```
915559/950029 = 0.963716897063
```

HMM Viterbi – test (ass1-tagger-test)



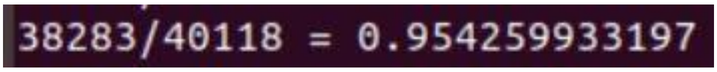
```
38027/40118 = 0.947878757665
```

HMM Greedy – test (ass1-tagger-test)



```
38223/40118=0.952764345182
```

MEMM – test (ass1-tagger-test)

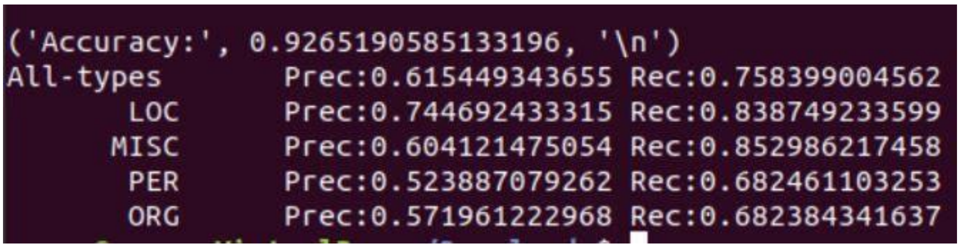


```
38283/40118 = 0.954259933197
```

NER

We used the following formula: $$F_\beta = (1+\beta^2)\frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}$$

HMM Greedy

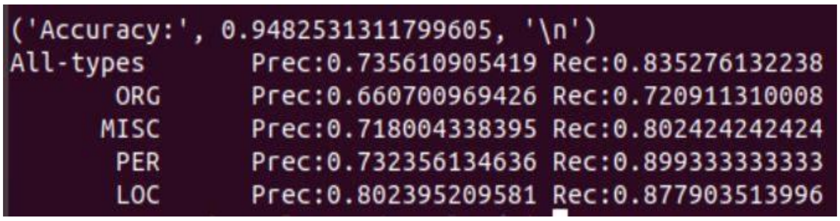


```
('Accuracy:', 0.9265190585133196, '\n')
All-types        Prec:0.615449343655 Rec:0.758399004562
       LOC       Prec:0.744692433315 Rec:0.838749233599
      MISC       Prec:0.604121475054 Rec:0.852986217458
       PER       Prec:0.523887079262 Rec:0.682461103253
       ORG       Prec:0.571961222968 Rec:0.682384341637
```

Accuracy: 0.92651

All-types:	Prec: 0.61544	Rec: 0.75839	F1: 0.67947
ORG:	Prec: 0.74469	Rec: 0.83874	F1: 0.78892
MISC:	Prec: 0.60412	Rec: 0.85298	F1: 0.70729
PER:	Prec: 0.52388	Rec: 0.68246	F1: 0.59274
LOC:	Prec: 0.57196	Rec: 0.68238	F1: 0.62230

HMM Viterbi



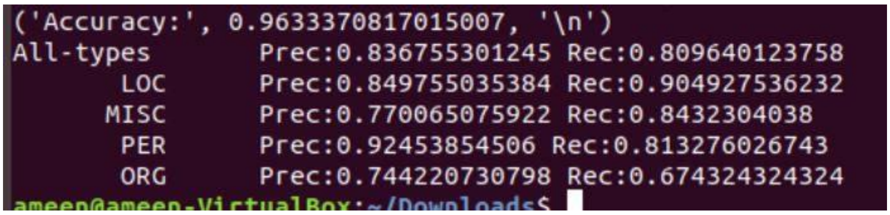
```
('Accuracy:', 0.9482531311799605, '\n')
All-types        Prec:0.735610905419 Rec:0.835276132238
      ORG        Prec:0.660700969426 Rec:0.720911310008
     MISC        Prec:0.718004338395 Rec:0.802424242424
      PER        Prec:0.732356134636 Rec:0.899333333333
      LOC        Prec:0.802395209581 Rec:0.877903513996
```

Accuracy: 0.948253

All-types:	Prec: 0.73561	Rec: 0.83527	F1: 0.78227

ORG:	Prec: 0.66070	Rec: 0.72091	F1: 0.68949
MISC:	Prec: 0.71800	Rec: 0.80242	F1: 0.75786
PER:	Prec: 0.73235	Rec: 0.89933	F1: 0.80729
LOC:	Prec: 0.80239	Rec: 0.87790	F1: 0.83844

Maxest Greedy

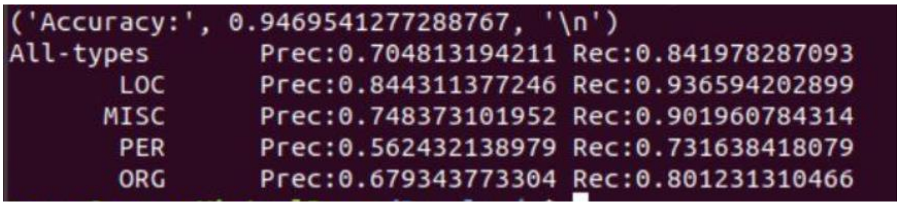


```
('Accuracy:', 0.9633370817015007, '\n')
All-types       Prec:0.836755301245 Rec:0.809640123758
       LOC      Prec:0.849755035384 Rec:0.904927536232
      MISC      Prec:0.770065075922 Rec:0.8432304038
       PER      Prec:0.92453854506 Rec:0.813276026743
       ORG      Prec:0.744220730798 Rec:0.674324324324
ameen@ameen-VirtualBox:~/Downloads$
```

Accuracy: 0.96333

All-types:	Prec: 0.83675	Rec: 0.80964	F1: 0.82297
ORG:	Prec: 0.84975	Rec: 0.90492	F1: 0.87646
MISC:	Prec: 0.77006	Rec: 0.84323	F1: 0.80498
PER:	Prec: 0.92453	Rec: 0.81327	F1: 0.86533
LOC:	Prec: 0.74422	Rec: 0.67432	F1: 0.70754

MEMM Viterbi



```
('Accuracy:', 0.9469541277288767, '\n')
All-types       Prec:0.704813194211 Rec:0.841978287093
       LOC      Prec:0.844311377246 Rec:0.936594202899
      MISC      Prec:0.748373101952 Rec:0.901960784314
       PER      Prec:0.562432138979 Rec:0.731638418079
       ORG      Prec:0.679343773304 Rec:0.801231310466
```

Accuracy: 0.94695

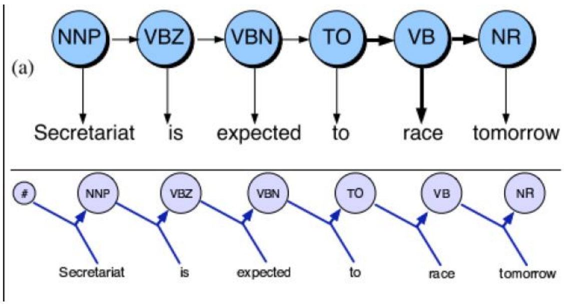
All-types:	Prec: 0.70481	Rec: 0.84197	F1: 0.76730
ORG:	Prec: 0.84431	Rec: 0.93659	F1: 0.88805
MISC:	Prec: 0.74837	Rec: 0.90196	F1: 0.81801
PER:	Prec: 0.56243	Rec: 0.73163	F1: 0.63596
LOC:	Prec: 0.67934	Rec: 0.80123	F1: 0.73526

4. Is there a difference in behavior between the hmm and maxent taggers? discuss.

There is a difference in the behavior of the two taggers.
The HMM is a model for the joint distribution of states (POS tags) and observations (words) and is captured by the probability for state to state (follows the Markov assumption) and the probability for state to observation. In a nutshell, the transitions between states are restricted to the immediate past.

In contrast, the MaxEnt tagger is a multinomial logistic regression for classification. It integrates feature information from the observations in addition to the previous state knowledge to derive more accuracy. Useful features can help improve it and they can be incorporated easily.

This figure shows the difference between the models:



(on top, hidden Markov model (HMM) and on the bottom, the Maximum Entropy model)

In addition, training the MaxEnt is different from the straight forward counting method used in HMM, therefore, it takes more time to train the data for the MaxEnt tagger.

5. Is there a difference in behavior between the datasets? discuss.

There is. The POS dataset contains a big number of tags, therefore, not all the possible sequences are captured and the assumption of the next tag based on previous tags is limited to the seen tags.
On the other hand, the NER dataset contains only a few tags, therefore, all the possible sequences are captured. We need to consider the word itself and its features more heavily than the previous words.

6. What will you change in the hmm tagger to improve accuracy on the named entities data?

We can use external knowledge to improve accuracy. For example,
lists extracted from the web that cover common names, countries, monetary units, temporal expressions, etc. While these gazetteers have excellent accuracy, they do not provide sufficient coverage. To further improve the coverage, we can extract 16 gazetteers from Wikipedia, which collectively contain over 1.5M entities for different gazetteers.

7. What will you change in the memm tagger to improve accuracy on the named entities data, on top of what you already did?

We can match against each gazetteer a weight as a separate feature in the system (this allows us to trust each gazetteer to a different degree).

8. Why are span scores lower than accuracy scores?

The span scores are based on a combination between the precision score and the recall score. Whereas the accuracy scores are more like the precision score.