Computational Linguistics

LIN 467/567 – Spring 2018

Homework 3

Due date: end of March 23rd

Instructions

Note: **Answers not submitted as specified below will not be graded**.

Upload to UBLearns a single zip file containing your answers to the two questions below. The zip file must be named '`hw3UBitNAME.zip`, where UBitNAME is your UB email's name. Don't forget to click the button after uploading the file, or it will not be submitted.

Students can work with others but must ultimately create their own individual answers. Any homework assignments that are sufficiently similar may be considered instances of plagiarism, and will be penalized and reported. See syllabus for more information about UB's official policy on ethical behavior.

Exercises

1. There is a total of twelve logically possible taggings for the sentence *Kids like sprouts* ('Kids' = NN / VB, 'like' = VB / PREP / ADJ, 'sprouts' = NN / VB). Construct the (maxed) trellis chart in a spreadsheet (Excel, OpenOffice, or GoogleDocs, for example) and determine what is the most likely tag sequence for that string given the HMM below. Your answer must be submitted as a PDF file called UBitNAME.pdf.

   [3 points]

   σ(NN,VB) = 0.6
   σ(@,VB) = 0.4
   σ(NN,PREP) = 0.3
   σ(@,NN) = 0.5
   σ(VB,VB) = 0.15
   σ(VB,PREP) = 0.25
   σ(VB,ADJ) = 0.3
   σ(VB,NN) = 0.6
   σ(NN,ADJ) = 0.01
   σ(VB,@) = 0.4
   σ(PREP,VB) = 0.5
   σ(PREP,NN) = 0.3
   σ(ADJ,VB) = 0.008
   σ(ADJ,NN) = 0.4
   σ(NN,@) = 0.6

   τ(kids,NN) = 0.4
   τ(kids,VB) = 0.1
   τ(like,VB) = 0.7
   τ(like,ADJ) = 0.005
   τ(sprouts,NN) = 0.4
   τ(sprouts,VB) = 0.2
   τ(like,PREP) = 0.02

2. The file viterbi.pl is a Prolog implementation of a modified Viterbi algorithm which lists all possible taggings and their probabilities instead of simply computing the best one. This is useful to determine how ambiguous the input is, how uncertain the tagger is, or if there is a need to backtrack to second most probable tagging, and so on. However, viterbi.pl is incomplete: it does not have the HMM model itself. To complete this exercise, you must:

   (a) Tag the words in the DA.txt mini corpus and create a tagged version called TDA.txt. Use the StanfordNLP suite MaxEnt tagger, by issuing the following (single) command:[1]

   ```
   java -cp "*" edu.stanford.nlp.tagger.maxent.MaxentTagger -model
   models/english-left3words-distsim.tagger -textFile DA.txt -tokenize true
   -outputFormat tsv -outputFile TDA.txt
   ```

   Before you can run the above, you must download the MaxEnt tagger and JKD installed (you may already have it, depending on your operative system).

   (b) Use bash commands to extract a Prolog-readable HMM from your TDA.txt and add it to viterbi.pl. Do not smooth the model.

   Your answer should therefore consist of (i) the TDA.txt tagged corpus created from a normalized version of DA.txt, (ii) a text file bashUBitNAME.txt containing bash commands used to extract a Prolog-readable HMM from the tagged corpus TDA.txt, and (iii) an augmented `viterbi.pl` text file named hmmUBitNAME.pl that now contains the HMM your derived from TDA.txt. The `viterbi.pl` file must successfully run the Viterbi algorithm over word sequences (numbers and tags are fictitious):[2]

```
?- viterbi_all([he,can,can,a,can],X).
X = tag(2.5515e-12, [pron, aux, v, det, aux]) ;
X = tag(1.27575e-12, [pron, aux, v, det, v]) ;
X = tag(1.96466e-11, [pron, aux, v, det, n]);
false
```

[3 points]

[1] To read more go to https://nlp.stanford.edu/software/tagger.shtml. This tagger used the Penn Treebank Tagset (see J&M'08 for more discussion).

[2] The number 2.5515e-12 means $2.5515 \times 10^{-12} = 0.0000000000025515$. This is the underflow problem discussed in class, caused by multiplying probabilities. If you wish, modify viterbi.pl so that it works in log-space instead.