Advanced Methods in NLP– Spring 2017
Assignment #3

Idan Rejwan (204085948)
Avi Caciularu (203056585)

April 2017

Question 1

a) Implemented in code.

Question 2

a) Implemented in code.

b) We get 91.5% accuracy for the dev set.

Question 3

a) Implemented in code.

b) We've improved our run-time complexity by going only over the tags which gave a positive emission probability (not 0).

c) As in the previous exercise, we performed grid search over λs combinations and got that:
The combination $\lambda_1 = 0.6, \lambda_2 = 0.3, \lambda_3 = 1 - \lambda_2 - \lambda_1 = 0.1$ gives the best results: 95% accuracy for the dev set.

Question 3

a) Implemented in code.

b) Implemented in code.

c) We've improved the run-time complexity using beam-search: in each iteration we saved only the top 100 probabilities of word tuples u, v, and in the next iteration we ran only on those 100 words as last two words. This way, we calculate only 100×45 probabilities in each step, instead of 45^3 - 20boosting of factor 20.

d) We get 95.26% accuracy for the dev set when using greedy decoding, and 95.32% accuracy for the dev set when using Viterbi decoding.

e) Since all the models accuracy were about the same 95%, we preferred to analyze the errors of the HMM model, because this model is interpretable from a generative perspective. We found a few kinds of errors:

   (a) Rare words - many of the errors were related to rare words which were replaced by "UNK" token. For example:

   - "He adds, 'This isn't 1987 UNK.'"
     True label for UNK: VBN
     Model's prediction: NNS
   - "The above represents a triumph of either UNK or UNK.
     True label for both UNK: NN
     Model's prediction for both: NNP

   Even refining it into sub-categories of rare words doesn't solve the problem. For example, we replaced every capitalized word with the token "CAP", since those words would probably represent proper nouns. However, we found the following sentence:

   - "CAP CAP:"
     True labels: NN NNS
     Model's predictions: NNP NNP

   The reason for this problem is that the replaced tokens for rare words are very diverse when considering POS. Namely, the number of POS tags that can possibly emit the token "UNK" (and other rare words tokens) is very big in comparison to "regular" words. For example, the token "UNK" has 31 possible tags (out of 45), where the word "the" has only 5.

(b) Dominant emission probability - there are cases when the emission probability for a given word by some tag is so dominant, that it "dominates" the model's prediction and biases it to predict only by the emission probability without considering the transition probability. For example:

- "That's baseball."
  True label for suffix 's: VBZ
  Model's prediction: POS

In the given example, the probability for "POS" to emit 's is 93%, whereas for "VBZ" it's 5.6%. For that reason, even though possessive is not common after the word "that", the model is much more affected by the emssion probability than the transition probability.

(c) Unclear discrimination (or: human errors) - the discrimination between some POS tags, for example JJ and VBN, is not clear, and it seems that human labeling is not always consistent:

- "The complicated language in the huge new law has UNK the fight."
  True label for word "complicated": VBN
  Model's prediction: JJ
- "UNK was subdued as most UNK watched the latest market statistics on television."
  True label for word "complicated": JJ
  Model's prediction: VBN