ANLP Tutorial Exercise Set 2 (for tutorial groups in week 4)

v1.4
School of Informatics, University of Edinburgh
Sharon Goldwater

This week's tutorial exercises focus on HMMs and tagging.

Exercise 1.
Suppose we want to train an HMM tagger for the task of Named Entity Recognition (NER). We are interested in only two kinds of named entities: persons (PER) and organizations (ORG), which include corporate and political entities. We have the following training data:[1]

> David/PER William/PER Donald/PER Cameron/PER (born 9 October 1966) is a British politician who has served as the Prime Minister of the United/ORG Kingdom/ORG since 2010 , as Leader of the Conservative/ORG Party/ORG since 2005 and as the Member of Parliament for Witney/ORG since 2001 .
>
> Cameron/PER studied Philosophy , Politics and Economics at Brasenose/ORG College/ORG , Oxford/ORG .
>
> He then joined the Conservative/ORG Research/ORG Department/ORG and became special adviser , first to Norman/PER Lamont/PER and then to Michael/PER Howard/PER .
>
> He was Director of Corporate Affairs at Carlton/ORG Communications/ORG for seven years .
>
> Cameron/PER first stood for Parliament in Stafford in 1997 .

In this data we show only the tags for the tokens belonging to the person and organization categories. Assume all other tokens have the tag OTH, which is not shown. Tokens are separated by whitespace. *Note:* There are a total of 73 types and 105 tokens in the text.

a) Give the transition probability matrix estimated from this training data using maximum-likelihood estimation. Don't forget to include beginning and end of sentence markers.

b) Now do the same but using add-one smoothing. Assume that all sentences must contain at least one word (i.e., $P(\texttt{</s>}|\texttt{<s>})$ is zero even in the smoothed model).

c) Again using add-one smoothing, what are the estimates for $P(\text{Cameron}|\texttt{PER})$ and $P(\text{Cameron}|\texttt{ORG})$?

Exercise 2.
This question also deals with NER and HMMs, but asks you to consider the nature of the problem and proposed solution, rather than working through the mathematical details.

a) Suppose we used *four* tags for this task: the three already mentioned, plus a LOC tag for locations. In a general text, will context always be able to disambiguate between the LOC and ORG tags? Justify your answer.

b) Can you think of any sources of information that might help an automatic NER system perform better, but which *are not* used by an HMM tagger? Back up your answer with examples from the text here, or give examples that could occur in another text.

[1] The text is a lightly edited version of the start of the Wikipedia article on David Cameron, downloaded Oct 2015.

c) POS taggers are normally evaluated using *accuracy*: (the percentage of the tags assigned by the tagger that agree with the gold standard). Does this evaluation measure also make sense for NER, or are there other evaluation measures we've discussed in class that you think would make more sense? Why?

Exercise 3.

Consider a simple HMM POS tagger with only five tags (plus the beginning and end of sentence markers, <s> and </s>). The transition probabilities for this HMM are given by the table on the left below, where cell [i,j] is the probability of transitioning from state i to j (i.e., $P(\text{state}_j|\text{state}_i)$). A subset of the output probabilities are given by the table on the right, where cell [i,j] is the probability of state i outputting word j (i.e., $P(\text{word}_j|\text{state}_i)$). We assume there are other possible output words not shown in the table, and that the <s> and </s> states output <s> and </s> words, respectively, with probability 1.

	CD	PRP	NN	VB	VBD	</s>
<s>	.5	.2	0	.3	0	0
CD	.2	0	.3	.2	.2	.1
PRP	.1	.1	0	.3	.4	.1
NN	.05	.15	.2	.25	.3	.05
VB	0	.2	.6	0	0	.2
VBD	0	.1	.6	0	0	.3

	one	cat	dog	bit	...
CD	.1	0	0	0	
PRP	.02	0	0	0	
NN	.05	.03	.04	.007	
VB	0	0	.03	0	
VBD	0	0	0	.06	

a) In the Penn Treebank tag scheme, what do the five different tags mean? Give example sentences illustrating the use of each word in the output matrix with each of its possible tags. (Your sentences should be real English, not limited to just the words/tags used in our tiny HMM.)

b) Using the HMM probability matrices, compute $P(\vec{w},\vec{q})$ (the joint probability of words and tags) for the sentence $\vec{w}$ = <s> one dog bit </s> with tags $\vec{q}$ = <s> CD NN NN </s>.

c) Now, hand-simulate the Viterbi algorithm in order to compute *highest probability* tag sequence $\vec{q}'$ for the given sentence, and the joint probability $P(\vec{q}',\vec{w})$, without enumerating all possible tag sequences. That is, fill in the cells in the following table, where cell $[j,t]$ should contain the Viterbi value for state j at time t, and you should also use backpointers to keep track of the best path. The rows of the table are already labeled with the different states, and the columns are already labeled with the observations at each time step.

Hint: For this particular HMM, a lot of the cells will have zeros in them. Try to work out ahead of time which these are, so you only need to do the Viterbi computations for the other cells.

	<s>	one	dog	bit	</s>
<s>					
CD					
PRP					
NN					
VB					
VBD					
</s>					

d) As you've seen, Viterbi probabilities get very small very fast. In practice, the algorithm is normally implemented using log probabilities to avoid underflow (as we did in the lab). The value in each cell is now a *negative log probability* (or *cost*), and we end up computing $-\log P(\vec{w},\vec{q})$. Work out what the equations need to be in this version of the algorithm. That is, what do we compute to get the value in cell (j,t)?