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

v1.3
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 $P(\text{Cameron}|\texttt{ORG})$?

Solution 1.

a) Note that punctuation are separate tokens, and tagged OTH, so all words before </s> are OTH. The probabilities are therefore

	PER	ORG	OTH	</s>
<s>	3/5	0	2/5	0
PER	5/10	0	5/10	0
ORG	0	6/13	7/13	0
OTH	2/82	7/82	68/82	5/82

b) These can be obtained from the MLE probability matrix as follows: in the <s> row, add 1 to each numerator and 3 to each denominator in the first three columns, leaving a 0 in the fourth column. (We added a 1 to each of three different options for the next tag, so we

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

need to add 3 in the denominator for a valid distribution.) In all other rows, add 1 to each numerator and 4 to each denominator. (Similar reasoning.)

c) MLE are 3/10 and 0/13, so we get 4/83 and 1/86. (In this case, we assume the vocabulary is just what we've seen in the corpus, so there are 73 different words, thus 73 possible outputs for each tag. Adding a count of one for each in the numerator means adding 73 in the denominator.)

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.

Solution 2.

a) Arguably, no: for example, I've tagged Witney here as an organization, as it's the political constituency (thus political entity, i.e., organization) that Cameron represents. But it is also a location, and this sentence doesn't make clear which meaning the writer intended. The same ambiguity holds for the other locations mentioned in the text. One might ask whether disambiguating these cases even matters, as the meaning of the sentence is basically the same either way. This is a good example of a case where humans don't normally notice (or care) about ambiguity, but it can make life difficult for corpus annotators and computer systems.

   In general, when you are justifying an argument, it is often a good idea to find concrete examples to support your point. This is especially effective if you can find a good counterexample. That is, if you want to argue that claim X does not hold in general, then a single good counterexample can prove that point. Similarly, if you are trying to argue that X is possible, an example will provide evidence for that claim. Examples are not sufficient to prove that a claim holds in general, but thinking about them might still give you something to say about the kinds of cases in which the claim is likely to be true.

b) Probably the most obvious type of information is features of the words themselves, for example capitalization. (It's clear from the provided text that capitalized words are much more likely to be NE's.) Other sources of information could be used, especially if outside resources are considered. For example, one could use a database of names.

   Note that the HMM already uses information such as: NEs often consist of multiple words in a row, and don't typically follow one another without a non-NE in between (encoded in the tag transition probabilities).

   POS information could also help—this information isn't currently used by the example tagger here, but of course this information can be obtained from an HMM POS tagger. You might want to consider how you would build an HMM for NER that also incorporates POS information (effectively, doing NER and POS tagging simultaneously).

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)?

Solution 3.

a) cardinal number, personal pronoun, singular/mass noun, base verb, past tense verb. Examples include:

- One/`CD` person is here. One/`PRP` can see Arthur's Seat from here. The one/`NN` you are looking for is here.
- I like the the cat/`NN`.
- The dog/`NN` is brown. His past mistakes dog/`VB` him to this day.
- Give me a bit/`NN` of cake. The dog bit/`VBD` me.

b) $(.5)(.3)(.2)(.05)(.1)(.04)(.007) = 4.2 \times 10^{-8}$. (In an exam, you would not be required to complete the multiplication, only write down the terms.)

c) The probabilities are as follows:

	`<s>`	`one`	`dog`	`bit`	`</s>`
`<s>`	1	0	0	0	0
`CD`	0	.05	0	0	0
`PRP`	0	.004	0	0	0
`NN`	0	0	6×10^{-4}	1.26×10^{-6}	0
`VB`	0	0	3×10^{-4}	0	0
`VBD`	0	0	0	1.08×10^{-5}	0
`</s>`	0	0	0	0	3.24×10^{-6}

If we number rows and columns starting from 0, then the backpointers are:

- in column `one`, both non-zero cells came from `<s>`.
- in column `dog`, both non-zero cells came from `CD`.
- in column `bit`, the `NN` cell came from `VB` and the `VBD` cell came from `NN`.
- in column `</s>`, the non-zero cell came from `VBD`.

Following the backpointers shows that the highest probability path is:

`<s> CD NN VBD </s>`

which is in this case the correct tag sequence.

d) The only difference is subtracting log probs instead of multiplying probs and taking a min instead of a max, because we're now using costs. For state j at time t, let $c(j,t)$ be the cost in cell $[j,t]$ of the chart. You'd need to compute:

$$\begin{aligned}
c(j,t) &= -\log v(j,t) && (1)\\
&= -\log \left[\max_{i=1}^{N} \left(v(i,t-1)\cdot a_{i,j}\cdot b_j(o_t)\right)\right] && (2)\\
&= -\max_{i=1}^{N} \log\left[v(i,t-1)\cdot a_{i,j}\cdot b_j(o_t)\right] && (3)\\
&= \min_{i=1}^{N} -\log\left[v(i,t-1)\cdot a_{i,j}\cdot b_j(o_t)\right] && (4)\\
&= \min_{i=1}^{N}\left[-\log v(i,t-1) - \log a_{i,j} - \log b_j(o_t)\right] && (5)\\
&= \min_{i=1}^{N}\left[c(i,t-1) - \log a_{i,j} - \log b_j(o_t)\right] && (6)
\end{aligned}$$