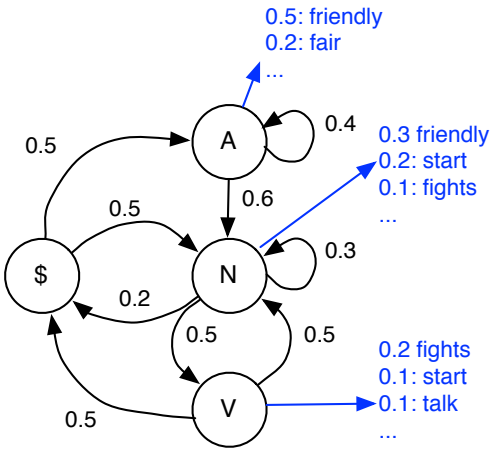
NLP1 2015-16: Assignment 2

The assignment considers two problems: a hidden Markov model (a toy PoS tagging problem) and parsing with context-free grammars (a basic CFG example and grammar refinement).

1 Hidden Markov model

Consider a hidden Markov model (HMM) represented by the following state diagram:



The state `$` is the start and end state of the HMM. The states `A`, `N` and `V` represent adjectives, nouns and verbs, respectively. This HMM can generate both simple sentences (e.g., *friendly fights start*) or just noun phrases (e.g., *friendly fights*). It can also generate some passages which are ungrammatical (e.g., *flights start fair talk*). Note that some words generated by the HMM are ambiguous: for example, *friendly* can be both an adjective or a noun (= a friendly game).

In this exercise we will consider an (ambiguous) passage: *friendly fights start*

1.1 Viterbi (10 points)

Use the Viterbi algorithm to compute the most likely sequence:

$$\arg \max_{s_1,\ldots,s_3} P(s_1, \ldots, s_3, x_1, \ldots, x_3 | A, B),$$

where A and B are the emission and transition parameters). Represent the Viterbi lattice either as a table (as in slide 102 of the lecture 4) or as a graph (as in slide 105 but with scores on top). Report the most likely sequence and its probability.

1.2 Forward and backward probabilities (10 points)

Compute forward and backward probabilities for all states and positions in the sentence. Present this information the same way as you did it for Viterbi (i.e. in a table or on top of a graph).

1.3 Posterior probabilities for a state (10 points)

Explain how posterior probabilities $p(s_t|x_1, \ldots, x_n)$ are computed using the backward and forward probabilities. Compute $p(s_3 = N|x_1, \ldots, x_n)$ and $p(s_3 = V|x_1, \ldots, x_n)$ (i.e. posterior probabilities of admissible PoS tags for the word *start* in the passage *friendly fights start*).

1.4 Maximum-a-posteriori (MAP) vs. maximum likelihood (10 points)

Consider the PoS tag which maximises the posterior for the word *start*: $\arg\max_{s\in\{N,A,V\}} p(s_3 = s|x_1, \ldots, x_n)$. Is this PoS tag the same as the one predicted by Viterbi for the word *start*?

If not, explain why it is not the same (hint: think about the alternative PoS tag sequences and their probabilities).

1.5 Which decoding to choose? (10 points)

The previous question suggested that there exist two alternative ways how to label a sequence of words:

1. Choosing the most likely sequence (as in Viterbi):

$$(s_1, \ldots, s_n) = \arg\max_{s_1,\ldots,s_3} P(s_1, \ldots, s_3, x_1, \ldots, x_3|A, B).$$

2. Labeling based on the posterior probabilities (aka Maximum a-posteriori decoding):

$$s_t = \arg\max_s p(s_t = s|x_1, \ldots, x_n),$$

   for every t.

Generally, not focusing on this specific HMM and the specific sentence, which strategy would you choose if you care about predicting the whole sequence of tags right? (i.e. you do not receive any credit for predicting partially correct sequences). Why?

Which strategy would you choose if you want to maximize per word accuracy (i.e. the number of correctly predicted tags divided by the sentence length). Why?

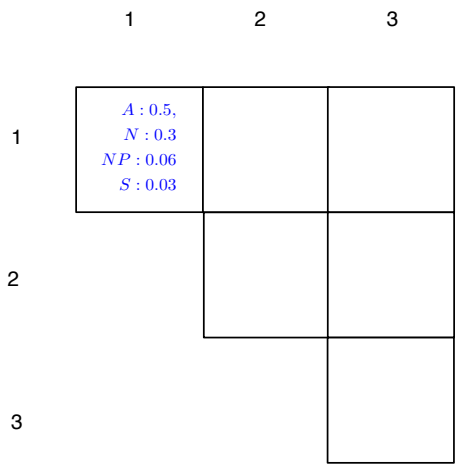
2 Inside-outside probabilities 20 points

We consider an (ambiguous) passage: *friendly fights start*

In this exercise, we focus on syntactic ambiguity rather than only on ambiguity in PoS tag assignment. Consider the following grammar:[1]

[1]Note that this is not a complete grammar: the sum of rule probabilties for pre-terminals A, N and V do not sum to 1.

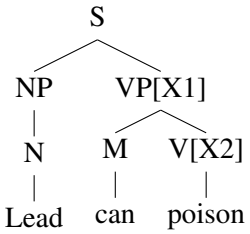
(1)	$S \rightarrow NP$	0.5
(2)	$S \rightarrow NP\ VP$	0.5
(3)	$NP \rightarrow N\ NP$	0.5
(4)	$NP \rightarrow A\ NP$	0.3
(5)	$NP \rightarrow N$	0.2
(6)	$VP \rightarrow V$	0.6
(7)	$VP \rightarrow V\ NP$	0.4
(8)	$A \rightarrow friendly$	0.5
(9)	$A \rightarrow fair$	0.2
(10)	$N \rightarrow friendly$	0.3
(11)	$N \rightarrow start$	0.2
(12)	$N \rightarrow fights$	0.1
(13)	$V \rightarrow fights$	0.2
(14)	$V \rightarrow start$	0.1
(15)	$V \rightarrow talk$	0.1
.....		



Fill in the chart (show on the right in the above figure) with **inside** probabilities.[2] What is the probability of the sentence according to the PCFG? Note that here we use both binary and unary inner rules (we considered only binary inner rules in the lecture about inside-outside probabilities), but you should be able to generalise the inside-outside algorithm to support unary rules in the same way we generalised CKY to support them.

3 Grammar refinement 30 points

Consider the following partially observable tree T:



Unlike the example in the lecture, here we split only two types of symbols: verb phrases (VP), and PoS tags for verbs (V). Each of them is split into 2 sub-categories. The example grammar is presented below. Intuitively, you can think that $VP[1]$ and $V[1]$ represent mostly transitive phrases and verbs (as in *John loves beer*), whereas $VP[2]$ and $V[2]$ exhibit preference towards intransitive ones (as in *John runs*). Show how the expected counts are computed in the EM-algorithm for the

[2]Note slight difference in notation used to refer to spans in the CKY and the Inside-Outside lectures. In this chart we use the one from the IO lectures (i.e. w_{12} refers to *friendly fights*).

given tree (**not** the full-blown inside-outside algorithm but its restricted version applied to grammar refinement). In other words, use a form of inside and outside probabilities to compute posteriors $P(S- > NP\ VP[X1]|T)$, $P(VP[X1] \rightarrow M\ V[X2]|T)$ and $P(VP[X2] \rightarrow poison|T)$ for all possible values of latent sub-categories (i.e. $X1,\ X2 \in \{0, 1\}$), where T is the partially observable tree shown above.

(1) $S \rightarrow NP\ VP[1]$	0.3
(2) $S \rightarrow NP\ VP[2]$	0.2
(3) $NP \rightarrow N$	0.1
(4) $VP[1] \rightarrow M\ V[1]$	0.01
(5) $VP[1] \rightarrow M\ V[2]$	0.01
(6) $VP[2] \rightarrow M\ V[1]$	0.02
(7) $VP[2] \rightarrow M\ V[2]$	0.2
(8) $N \rightarrow Lead$	0.1
(9) $M \rightarrow can$	0.2
(10) $V[1] \rightarrow poison$	0.01
(11) $V[2] \rightarrow poison$	0.1
.....	

4 Submission rules

The deadline for the assignment is Tuesday, November 24. This time, as the assignment requires some pictures, we are OK with you submitting a hand-written version. You can hand in your assignment during the lab session to a teaching assistant. Alternatively, you can submit a scanned or typed version over email to nlp.uva.2015@gmail.com with subject *Assignment 2*. In either case the deadline is at **13:00** on November 24.