Lab 4 Report: Viterbi and Beam Search

In this lab we will explore the implementation of a Sructured Perceptron for Named Entity Recognition. The perceptron needs to learn to predict one of 5 labels ['O', 'LOC', 'ORG', 'MISC', 'PER']. We will use the Viterbi and Beam Search algorithms to make this calculations faster than the previous lab.

Viterbi Algorithm

In order to implement the Viterbi algorithm, a matrix for each training instance is created of dimensions Y x N , where Y is the different labels, and N the words in the sentence.

For example, this would be the matrix for the sentence "San Francisco is great"

	San	**Francisco**	**is**	**great**
ORG	0	0	0	0
MISC	0	0	0	0
PER	2	1	0	0
LOC	4	2	0	0
O	0	0	1	1

In this particular case, the phi1 feature only consists of the current word and current tag, which means that there are no transition values between words or tags (total independence). This simplifies the algorithm a lot because the maximum score label is simply the maximum value for each column (in the above example would be "LOC LOC O O").

After 5 passes with shuffling we tested our weights and the result was a **micro F1 score of 0.73** which is the exact same result of the exhaustive search we did in the previous lab.

The difference in processing time is very surprising. In the previous lab each epoch took around 20 seconds to complete (100 seconds total), but with the Viterbi algorithm implemented, each epoch takes only 0.5 seconds for a total of 2.5 seconds. This is a reduction of almost 98% of training time!

To get the most weighted features, the results for each label were sorted by value and then by key, so they could be reproduced. This is because there could be more than 10 keys with the top "value" so each time the dictionary would give different results.

The most weighted features for each label were the following (and are the same as the previous lab)

ORG	**MISC**	**PER**	**LOC**	**O**
&	C$	A.	ABABA	"
ABC	162	A.de	ABDERDEEN	$
AD-DIYAR	AMERICAN	Aamir	ABIDJAN	&
AD	American	Adrian	ADDIS	'S
AHOLD	Argentine	Affleck	AIRES	'
AHRONOTH	Australian	Ahmed	AJACCIO	'm
AL-ANWAR	Austrian	Akam	AKRON	's
AL-WATAN	Baseball	Akram	AL-MUNTAR	(
AL	Bedi	Alan	ALKHAN-YURT	)
AN-NAHAR	Belgian	Alastair	ALLENTOWN	*

Beam Search

I was not able to implement Beam Search, but I will write my intuitions about the expected results, given what I learned about this algorithm. In my code I only applied a beam search of K=1, which is equal to greedy search (same result as Viterbi).

For this particular phi1 (current word - current label) the beam size should not affect accuracy, because it will always be capable of finding the maximum (only because there is no transition probabilities). So the top-k possible labels will always include the real label.

Usually the effect of the beam size on speed is inverse, so a bigger beam size will be slower than a smaller beam, because it has to evaluate more paths.

The effect of the beam size on accuracy is direct, a smaller (greedy search) beam size could be more inaccurate than a bigger one, because it might miss a more suitable label (not top-k).