STATISTICAL STRUCTURED PREDICTION
Question set 2

Ramon Ruiz Dolz

December 2018

1 Theoretical questions

The equation used to estimate a PCFG from a subset of derivations is the following one.

$$p(A \to \alpha) = \frac{\sum_{x \epsilon D} \frac{1}{P_\theta(x)} \sum_{t_x \epsilon x} N(A \to \alpha, t_x) * P_\theta(x, t_x)}{\sum_{x \epsilon D} \frac{1}{P_\theta(x)} \sum_{t_x \epsilon x} N(A, t_x) * P_\theta(x, t_x)} \tag{1}$$

1.1 Question 3

The estimation of the rule with the Inside-Outside algorithm with the example from the slides and the training set $D = \{$(la vieja)(demanda ayuda), la mujer oculta pelea, la vieja ayuda$\}$ is computed as follows,

$$P(Suj \to Art, Nom, Adj) = \frac{\frac{0.01176}{0.01266}}{\frac{0.0009}{0.0009} + \frac{0.0009+0.01176}{0.01266} + \frac{0.007}{0.007}} = 0.3096 \tag{2}$$

1.2 Question 4

In the same way as the previous question, the estimation of the rule with Viterbi algorithm and the same dataset D is computed as follows,

$$P(Suj \to Art, Nom, Adj) = \frac{\frac{0.01176}{0.01176}}{\frac{0.0009}{0.0009} + \frac{0.0009+0.01176}{0.01266} + \frac{0.007}{0.007}} = 0.3333 \tag{3}$$

The main difference between both algorithms is that while I-O takes into account all the possible trees, Viterbi only takes into account the maximum probability tree.

1.3 Question 5

This question asks to estimate the rule ($Suj \to Art, Nom, Adj$) with the Inside-Outisde algorithm and the training set $D = \{$la vieja demanda ayuda, la mujer oculta pelea, la vieja mujer oculta demanda ayuda$\}$. In order to work with this new training set we have defined a new rule like ($Suj \to Art, Adj, Nom, Adj$) with the probability of 0.2 and the probabilities of rules ($Suj \to Art, Nom$) and ($Suj \to Art, Adj, Nom$) have been decreased by 0.1. An image of the tree generated can be seen in Figure 1. After applying Inside-Outside, the new estimation for the rule obtained is the following one.

$$P(Suj \to Art, Nom, Adj) = \frac{\frac{0.01176}{0.01248}}{\frac{0.00072+0.00168}{0.0024} + \frac{0.00072+0.01176}{0.01248} + \frac{0.0002268}{0.0002268}} = 0.3141 \tag{4}$$

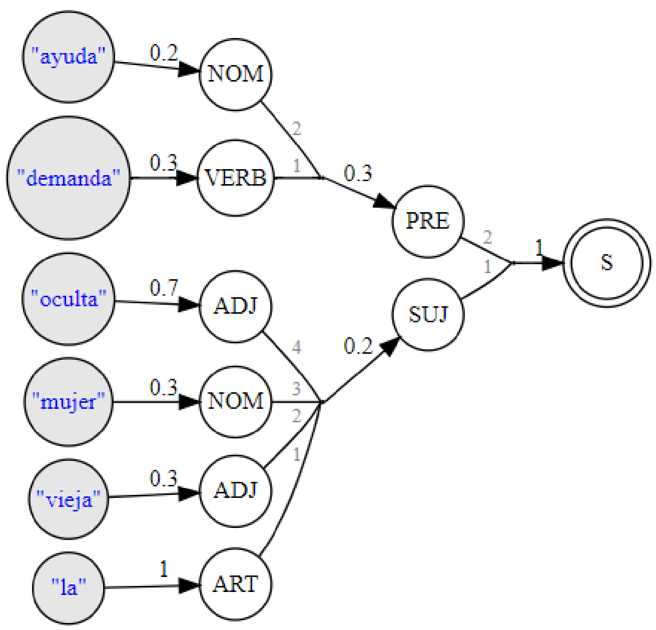

Figure 1: Tree generated with the new training sample

1.4 Question 6

On this question we are asked to use the k-best estimation algorithm in order to compute the same instance of the last question. K-best algorithm is a general way to estimate the rules. Viterbi and Inside-Outside algorithms are concrete instances of k-best algorithm. When k=1 the algorithm is equivalent to Viterbi and when k=max being max the total number of trees existing, the algorithm is equivalent to Inside-Outside. In this case, k=2 and the maximum number of trees existing are 2, so the result will be the same as computed in question 5.

$$P(Suj \to Art, Nom, Adj) = \frac{\frac{0.01176}{0.01248}}{\frac{0.00072+0.00168}{0.0024} + \frac{0.00072+0.01176}{0.01248} + \frac{0.0002268}{0.0002268}} = 0.3141 \tag{5}$$

2 Practical assignments

On this second practical assignment the main objective is to estimate grammars using the scfg toolkit. This assignment is divided in three activities. The two first activities purpose is to compare the results of estimating using Inside-Outside or estimating using Viterbi. The third exercise proposes a more complex grammar to represent geometric figures, and I will analyse the behaviour of this grammar when modifying the number of non terminal symbols.

2.1 Question 8

On this first part of the assignment, I have trained the four grammars given to us with the Inside-Outside algorithm. In addition the training has been done with a dataset without brackets and another with brackets. In the following tables it is possible to see the different values obtained in the structural and the statistical evaluation.

Grammar	Dataset	Log-likelihood	Perplexity
G1	Normal	-9193.20	9830.14
G1	Brackets	-5964.08	389.19
G2	Normal	-9306.65	11011.04
G2	Brackets	-9381.82	11870.60
G3	Normal	-5931.56	376.74
G3	Brackets	-5936.41	378.57
G4	Normal	-5931.91	376.87
G4	Brackets	-5936.26	378.51

Table 1: Resutls of the statistical evaluation of the grammars trained using I-O algorithm

Grammar	Dataset	Palindromes	APM
G1	Normal	364	27.03
G1	Brackets	627	57.12
G2	Normal	244	21.27
G2	Brackets	229	19.33
G3	Normal	999	27.41
G3	Brackets	999	25.81
G4	Normal	999	28.31
G4	Brackets	999	29.77

Table 2: Resutls of the structural evaluation of the grammars trained using I-O algorithm

2.2 Question 9

On the same way as the latter exercise has been done, I have trained all four grammars but with Viterbi algorithm this time. There have been also used two kind of datasets, one bracketed and other without brackets. On the following tables the results obtained for both statistical and structural evaluations can be observed.

Grammar	Dataset	Log-likelihood	Perplexity
G1	Normal	-9334.73	11324.62
G1	Brackets	-5964.28	389.27
G2	Normal	-9385.06	11909.14
G2	Brackets	-9457.90	12809.07
G3	Normal	-7443.54	1708.79
G3	Brackets	-7446.17	1713.29
G4	Normal	-5938.82	379.48
G4	Brackets	-5940.84	380.25

Table 3: Resutls of the statistical evaluation of the grammars trained using Viterbi algorithm

Grammar	**Dataset**	**Palindromes**	**APM**
G1	Normal	339	26.84
G1	Brackets	627	57.12
G2	Normal	268	22.66
G2	Brackets	271	23.32
G3	Normal	650	35.75
G3	Brackets	646	36.69
G4	Normal	999	29.20
G4	Brackets	999	28.13

Table 4: Results of the structural evaluation of the grammars trained using Viterbi algorithm

2.3 Question 10

On this final practical exercise I have trained four different grammars that produce rectangle triangles. The purpose of this exercise is to observe the behaviour of the grammar by modifying the number of non-terminal symbols existent on the grammar. Concretely, I have generated 4 different grammars, each one with 5, 10, 15 and 20 non-terminal symbols. After generating the grammars, I have trained them all with 700 iterations over the dataset SampleTriangle-10K given to us. Finally, in order to evaluate the grammars i have generated 1000 strings from each grammar and checked from those 1000 strings how many are rectangle triangles. The results are shown in the following table.

# non-terminal symbols	**#rectangle triangles**
5	29
10	63
15	61
20	84

Although the results obtained with the grammars with 15 and 20 non-terminal symbols are much better, the training process costs more than four times the smaller grammars.