STATISTICAL STRUCTURED PREDICTION

Question set (Part 1-B)

January 2022

Administrative issues

- This part represents the fourth part of the final score of PEE.
- The score of this part consists of 85% of theoretical questions and 15% of practical assignments.
- You can choose as many exercises as you like but the maximum score is 10 out of 10. The same rule is applied to the practical questions.
- The score of each exercise will depend on the depth, quality and justification of the answer and the analysis and future extensions of the results both in the theoretical exercises and practical exercises.
- Please present the results in PDF documents and uplodad your document throug a poliformat task.
- Deadline for delivering the theoretical questions and the report of practical assignment: **21 Jan 2022**

Theoretical questions

Question 1 (2 point)

Analyze the behavior of the Inside-Outside estimation algorithm when the initial probabilities associated to the rules are equiprobable. Justify your comments with the example in page 50 from Part I.4 Model Parameter Estimation (P-I.4 from now on),

Question 2 (2 points)

Compute the estimation of the rule $(\mathrm{Suj} \rightarrow \mathrm{Art\ Nom})$ with the example in page 50 from P-I.4 with the Inside-Outside algorithm with the following training sample that includes bracketed samples: $\mathcal{D} = \{(la\ vieja)(demanda\ ayuda), la\ mujer\ oculta\ pelea, la\ vieja\ ayuda\}$.

Question 3 (1 point)

Repeat the previous exercise but using the Viterbi-Score estimation algorithm.

Question 4 (3 points)

Compute the estimation of the rule $(Suj \rightarrow Art\ Nom)$ with the example in page 49 from P-I.4 with the Inside-Outside algorithm with the following training sample: $\mathcal{D} = \{la\ vieja\ demanda\ ayuda,\ la\ mujer\ oculta\ pelea,\ la\ vieja\ mujer\ oculta\ demanda\ ayuda\}$. You can use the hyp toolkit[1] and you have to provide a plot for the last sentence similar to the ones in slide 50 of P-I.4. Compute just one iteration.

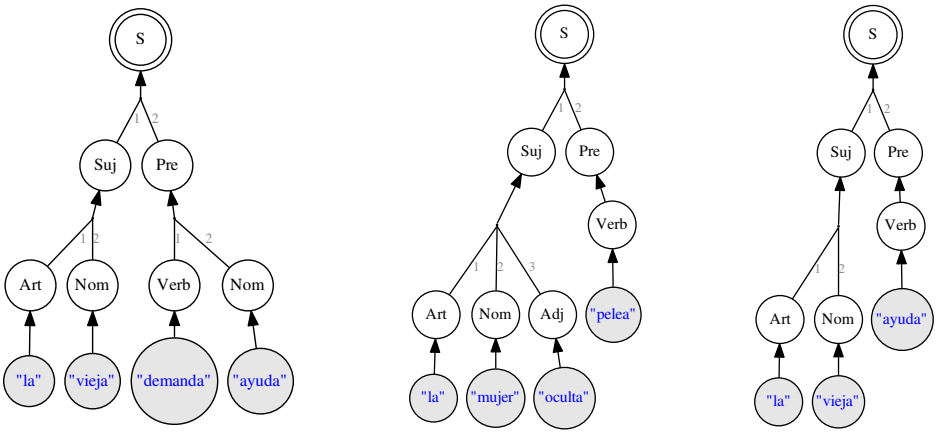
Question 5 (3 points)

Repeat the previous exercise but using the k-best estimation algorithm with $k = 2$. You can use the hyp toolkit. Compute just one iteration.

[1] https://github.com/sdl-research/hyp

Question 6 (4 points)

Apply expression (13) in slide 62 of P-I.4 to example of slides 50 and 51 to the rule $(Suj \rightarrow Art\ Nom)$. Consider that the following training samples that can be seen below. For each rule, consider the features as described in: *"Learning to Extract Information from Semi-structured Text Using a Discriminative Context-Free Grammar"*, P. Viola, M. Narasimhand, SIGIR 2005.[2]



[2] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.9192&rep=rep1&type=pdf

Practical questions

This practical part assumes that you are familiar with the scfg toolkit and with the triangle task. The toolkit includes a program for estimating a SCFG with the Inside-Outside algorithm, with the Viterbi-Score algorithm, with and without bracketed samples. For example, in the following commands:

```
scfg-toolkit/scfg_cgr -g MODELS/Gm1 -f MODELS/Gm1-new
scfg-toolkit/scfg_learn -g MODELS/Gm1-new -f MODELS/Gm1-new-100 -i 100 -m DATA/D
scfg-toolkit/scfg_learn -g MODELS/Gm1-new -f MODELS/Gm1-new-br-100 -i 100 -m DATA/Dpar
```

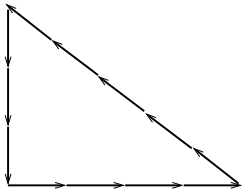
the first command creates a new SCFG, the second trains the grammar with the Inside-Outside algorithm with plain samples, and the third command trains the grammar with the Inside-Outside algorithm with bracketed samples.

Question 7 (5 points)

PCFG can be used to represent geometric figures. Consider the following primitives:

$$a = \nearrow,\quad b = \rightarrow,\quad c = \searrow,\quad d = \downarrow, e = \swarrow,\quad f = \leftarrow,\quad g = \nwarrow,\quad h = \uparrow .$$

Then, a rectangle triangle like this:



can be represented with the string $bbbbggggggddd$.

In the PCFG toolkit you will find a dataset `SampleTriangle-10K` representing rectangle triangles of this type. The samples are bracketed like this: $(bbbb)(ggggg)(ddd)$. A PCFG can be trained and tested as follows:

```
scfg-toolkit/scfg_cgr -g MODELS/G-triangle -f MODELS/Gt-0
scfg-toolkit/scfg_learn -g MODELS/Gt-0 -f MODELS/Gm1-new-700 -i 700 -m DATA/SampleTriangle-10K
scfg-toolkit/scfg gstr -g MODELS/G-triangle-20-V-10K -c 1000 > tri-test
awk -f scfg-toolkit/checkTriangle tri-test | grep Y | wc -l
```

The last command outputs the number of rectangle triangles in 1000 generated strings. The learning process can take some time. The number of non-terminal symbols in the PCFG is a critical point since the more non-terminal symbols the more flexibility the PCFG has to learn the samples, but it takes more time and more training samples are needed.

Complete the following table:

# non-terminal symbols	# rectangle triangles
5	
10	
15	
20	

Now we will work with three types of triangles: right triangles, equilateral triangles, and isoscalen triangles. These type of triangles exhibit some relations that PCFG may capture. For this purpose, bracketed samples can be used to represent the relations.

The following commands perform a simple experiment for givent training and test sets:

```
# train models
scfg-toolkit/scfg_learn -g MODELS/G-0 -f MODELS/right -m DATA/Tr-right -i 100
scfg-toolkit/scfg_learn -g MODELS/G-0 -f MODELS/equil -m DATA/Tr-equil -i 100
scfg-toolkit/scfg_learn -g MODELS/G-0 -f MODELS/isosc -m DATA/Tr-isosc -i 100
# classify with the trained models and get results
scfg-toolkit/scfg_prob -g MODELS/right -m DATA/Ts-right > r
scfg-toolkit/scfg_prob -g MODELS/equil -m DATA/Ts-right > e
scfg-toolkit/scfg_prob -g MODELS/isosc -m DATA/Ts-right > i
paste r e i | awk '{m=$1;argm="right"; if ($2>m) {m=$2;argm="equil";}
              if ($3>m) {m=$3;argm="isosc";}printf("right %s\n",argm);}' > results
scfg-toolkit/scfg_prob -g MODELS/right -m DATA/Ts-equil > r
scfg-toolkit/scfg_prob -g MODELS/equil -m DATA/Ts-equil > e
scfg-toolkit/scfg_prob -g MODELS/isosc -m DATA/Ts-equil > i
paste r e i | awk '{m=$2;argm="equil"; if ($1>m) {m=$1;argm="right";}
              if ($3>m) {m=$3;argm="isosc";} printf("equil %s\n",argm);}' >> results
scfg-toolkit/scfg_prob -g MODELS/right -m DATA/Ts-isosc > r
scfg-toolkit/scfg_prob -g MODELS/equil -m DATA/Ts-isosc > e
scfg-toolkit/scfg_prob -g MODELS/isosc -m DATA/Ts-isosc > i
paste r e i | awk '{m=$3;argm="isosc"; if ($1>m) {m=$1;argm="right";}
              if ($2>m) {m=$2;argm="equil";} printf("isosc %s\n",argm);}' >> results

cat results | scfg-toolkit/confus
#      equi isos righ  Err Err%
# equi  794  206    0  206 20.6
# isos  531  225  244  775 77.5
# righ  108  145  747  253 25.3

# Error: 1234/3000 = 41.13%
```

Question 8 (3 points)

Study the classification results depending on the algorithm used for training (Inside-Outside or Viterbi) and the type of samples (bracketed or not). Analyze the results.

Question 9 (3 points)

Study the classification results depending on the algorithm used for learning the PCFG and the size of the training data. You can generate more training data as follows:

```
scfg-toolkit/genFig -F 0 -c 1000  -l 2 -L 10 > DATA/Tr-right
```

You have to use a different seed for each training set in order to avoid a repeated sequence of training samples. Analyze the results.

Question 10 (10 points)

Implement the structured perceptron and compare the algorithm with the Inside-Outside and Viterbi algorithms in the triangle classification task.