ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE – LAUSANNE
POLITECNICO FEDERALE – LOSANNA
SWISS FEDERAL INSTITUTE OF TECHNOLOGY – LAUSANNE

Faculté Informatique et Communication
Introduction to Natural Language Processing (CS–431)
Chappelier, J.-C. & Rajman, M.

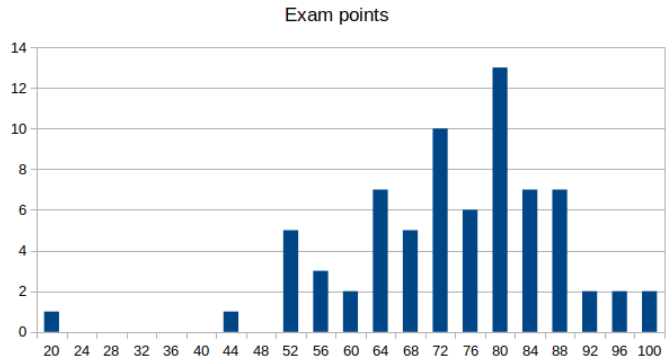
INTRODUCTION TO NATURAL LANGUAGE PROCESSING (CS–431)

2021 — Solution of the exam

Wednesday, January 20th, 2021.

Here are some statistics about this exam.

Distribution of points (min = 18.75, max = 97.75, avg = 72.04, stdev = 13.65, median = 73.21):



Statistics for questions I and III:

	I-1	I-2	I-3	I-Total	III-1	III-2	III-Total
min	0	0	0	1.5	0	0	0
max	4	3	2	9	2	5	7
avg	2.70	0.82	1.59	5.11	0.93	1.29	2.22
stdev	1.07	0.91	0.66	1.75	0.94	1.85	2.48
median	3	1	2	5.16	1	0	1

Statistics for question II:

	2.1 2	2.2 5	2.3 2	2.4 4	2.5 10	2.6 12	2.7 5	2.8 5	Total 45
min	0	0	0.5	0	0	0	0	0	2.25
max	2	5	2.5	4	10	12	5	5	45
avg	1.71	4.20	1.81	3.78	8.55	10.27	3.58	1.79	35.69
stdev	0.71	0.93	0.36	0.81	2.85	2.50	2.18	1.84	6.89
median	2	4.4	2	4	10	11.35	5	1.5	36.35

Statistics for question IV to VII:

	Total	Q4.1	Q4.2	Q4.3	Q4.4	Q4.5	Q4.6	Q4	Q5.1	Q5.2	Q5	Q6.1	Q6.2	Q6.3	Q6.4	Q6	Q7.1	Q7.2	Q7.3	Q7.4	Q7
min	11	0	0	0	0	0	0	1.50	0	0	0.00	0	0	0	0	0.00	0	0	0	0	0.00
max	42.25	1	1	1	4	2	7	16.00	1.75	2	3.75	2	5	2	5.5	13.5	6	2	3	2	13.00
avg	29.03	0.98	0.84	0.88	1.82	1.66	4.18	10.37	1.27	1.26	2.53	1.48	2.54	1.23	3.24	8.49	4.71	0.98	1.34	0.63	7.65
stdev	7.14	0.13	0.27	0.29	1.19	0.65	2.06	3.37	0.47	0.70	0.98	0.71	1.41	0.56	1.24	3.07	1.18	0.71	1.34	0.94	2.82
median	30	1	1	1	2	2	4.5	11.00	1.5	1.5	2.75	1.75	2.5	1.5	3.5	9.25	5	1	1	0	8.00

QUESTION I : n-grams [10 pt]

① **[4 pt]** Using a 3-gram model of characters, what is the expression of the ratio $\lambda = P(\text{address})/P(\text{dressage})$?
Provide the answer in the form of the *simplest* possible formulas (= with the fewer terms), using only model parameters.

$$\begin{aligned} P(\text{address}) =& P(\text{add}) \cdot P(\text{ddr}) \cdot P(\text{dre}) \cdot P(\text{res}) \cdot P(\text{ess}) \\ & / P(\text{dd}) \cdot P(\text{dr}) \cdot P(\text{re}) \cdot P(\text{es}) \\ P(\text{dressage}) =& P(\text{dre}) \cdot P(\text{res}) \cdot P(\text{ess}) \cdot P(\text{ssa}) \cdot P(\text{sag}) \cdot P(\text{age}) \\ & / P(\text{re}) \cdot P(\text{es}) \cdot P(\text{ss}) \cdot P(\text{sa}) \cdot P(\text{ag}) \end{aligned}$$

where $P(xy) = \sum_z P(xyz)$. Thus

$$\lambda = \frac{P(\text{add}) \cdot P(\text{ddr}) \cdot P(\text{ss}) \cdot P(\text{sa}) \cdot P(\text{ag})}{P(\text{dd}) \cdot P(\text{dr}) \cdot P(\text{ssa}) \cdot P(\text{sag}) \cdot P(\text{age})}$$

Notice that the blue part is simply $P(\text{dress})$ (which could also be written directly).

② **[4 pt]** How does the former ratio λ change if we introduce a word-boundary marker (as a new character)?
Explain/Comment the impact of this change.

So λ is now: $\lambda = P(\$\text{address}\$)/P(\$\text{dressage}\$)$, where we use '$' to denote word boundaries.

$$\begin{aligned} P(\$\text{address}\$) =& P(\$\text{ad}) \cdot P(\text{add}) \cdot P(\text{ddr}) \cdot P(\text{dre}) \cdot P(\text{res}) \cdot P(\text{ess}) \cdot P(\text{ss}\$) \\ & / P(\text{ad}) \cdot P(\text{dd}) \cdot P(\text{dr}) \cdot P(\text{re}) \cdot P(\text{es}) \cdot P(\text{ss}) \\ P(\text{dressage}) =& P(\$\text{dr}) \cdot P(\text{dre}) \cdot P(\text{res}) \cdot P(\text{ess}) \cdot P(\text{ssa}) \cdot P(\text{sag}) \cdot P(\text{age}) \cdot P(\text{ge}\$) \\ & / P(\text{dr}) \cdot P(\text{re}) \cdot P(\text{es}) \cdot P(\text{ss}) \cdot P(\text{sa}) \cdot P(\text{ag}) \cdot P(\text{ge}) \end{aligned}$$

Thus here λ becomes:

$$\begin{aligned} \lambda &= \frac{P(\$\text{ad}) \cdot P(\text{add}) \cdot P(\text{ddr}) \cdot P(\text{ss}\$) \cdot P(\text{sa}) \cdot P(\text{ag}) \cdot P(\text{ge})}{P(\text{ad}) \cdot P(\text{dd}) \cdot P(\$\text{dr}) \cdot P(\text{ssa}) \cdot P(\text{sag}) \cdot P(\text{age}) \cdot P(\text{ge}\$)} \\ &= \frac{P(\$\text{ad})}{P(\text{ad})} \cdot \frac{P(\text{ss}\$)}{P(\text{ss})} \cdot \frac{P(\text{dr})}{P(\$\text{dr})} \cdot \frac{P(\text{ge})}{P(\text{ge}\$)} \cdot \lambda_0 \end{aligned}$$

which contains more parameters, thus more chance to have better discrimination; typically the position of "dr(e)ss" within the word is better taken into accout.

③ **[2 pt]** Consider a 4-gram character model over an alphabet of 184 characters, which is estimated over a corpus of $1'283'267$ occurrences of words containing $8'136'785$ occurrences of 4-grams. Using a Dirichlet prior with a uniform parameter set to 0.002, what is the estimated value of a parameter associated to a 4-gram that appears only 3 times in the corpus?
Provide your answer in the form of a formula containing only numerical values.

$$(3 + 0.002) \,/\, (8'136'785 \;+\; 184^4 \times 0.002)$$

QUESTION II : Parsing [45 pt]

Consider the following SCFG (and lexicon):

```
S   -> NP VP        (p1)
S   -> NP VP PNP    (0.6)
NP  -> Adj NP       (0.3)
NP  -> N            (0.1)
NP  -> Det N        (p5)
NP  -> NP NP        (0.2)
NP  -> NP PNP       (p7)
VP  -> V NP         (p8)
VP  -> V NP PNP     (0.7)
PNP -> Prep NP      (p10)
```

```
farmers     N       q1
field       N       0.4
field       V       0.06
good        Adj     0.8
good        N       0.1
ground      Adj     q6
ground      N       q7
ground      V       q8
the         Det     q9
to          Prep    q10
till        Prep    0.05
till        V       0.7
```

where `p1`, `p5`, `p7`, `p8`, `p10`, `q1`, `q6`, `q7`, `q8`, `q9` and `q10` are (non-null) numbers between 0 and 1; and consider the following sentence:

`good field farmers till the ground`

① **[2 pt]** How many possible Part-of-Speech taggings does the above sentence have (with the above lexicon)? Briefly justify your answer.

24: $2 \times 2 \times 1 \times 2 \times 1 \times 3$

② **[5 pt]** Explain the differences between Part-of-Speech tagging and Context-Free parsing, in terms of task description, input, output and required resources.

	PoS TAGGING	Context-Free PARSING
task	assign to every input word its corresponding Part-of-Speech (syntactic role) within the context of the sentence	either syntactically recognize (decide whether the input sentence is syntactically correct or not) or syntactically analyze (give all parse trees)
input	sentence(s)/sequence of words	same!
output	tagged text (1 PoS to every word)	either boolean value (recognizer) or parse forest (set of syntactic trees)
resources	lexicon and parameters (HMM: probabilities)	lexicon and CF grammar
	or (in both cases) a corpus to learn them	

③ **[2 pt]** Propose some values for `p1`, `p5`, `p7`, `p8`, and `p10`, for the grammar to be a correct SCFG.

$$p_1 = 0.4, \quad p_5 + p_7 = 0.4, \quad p_8 = 0.3, \quad p_{10} = 1$$

I don't know why so many students had $p_5 + p_7$ wrong (having $p_5 + p_7 = 0.7$ or 0.8).

On the next page, you will find six propositions of parse trees for the sentence

`good field farmers till the ground`

④ **[4 pt]** Which of those trees are indeed correct parse trees according the above grammar? Briefly explain your answer.

T_2 and T_5 are wrong, all the other are correct.

T_2 is wrong because (for instance) there is no rule `NP -> Adj`.

T_5 is wrong because there is no rule `NP -> N Prep NP`.

All the others contain only valid rules.

⑤ **[10 pt]** Among the proposed correct parse trees, which is the most probable one (**2 pt**)? Justify your answer (**8 pt**; you can answer on/annotate the next page).

(see annotations on the next pages.)

$p(T_1) < p(T_3)$ since $p_2 \cdot p_8 < p_1 \cdot p_9$ ($0.6 \times 0.3 < 0.4 \times 0.7$), all the rest being the same).

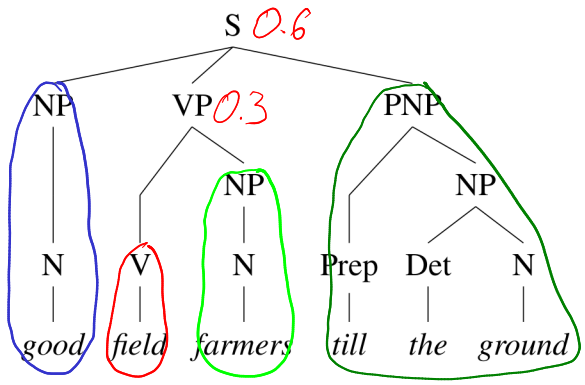
$p(T_4) < p(T_6)$ since $p_6 \cdot p_4 \cdot q_5 < p_3 \cdot q_4$ ($0.2 \times 0.1 \times 0.1 < 0.3 \times 0.8$), all the rest being the same.

So we're left with comparing T_3 and T_6.

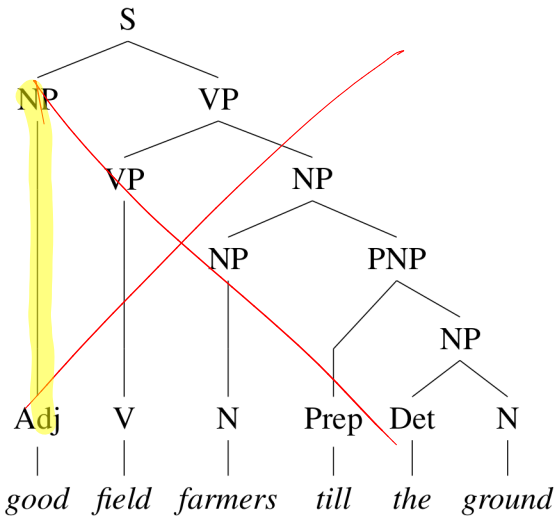
$p(T_3) < p(T_6)$ since $q_5 \cdot p_9 \cdot q_3 \cdot p_{10} \cdot q_{11} < p_3 \cdot q_4 \cdot p_6 \cdot q_2 \cdot p_8 \cdot q_{12}$ ($0.1 \times 0.7 \times 0.06 \times 1 \times 0.05 < 0.3 \times 0.8 \times 0.2 \times 0.4 \times 0.3 \times 0.7$, essentially because 0.06×0.05 is much smaller than $0.8 \times 0.2 \times 0.4$ (1 order of magnitude)), all the rest being the same.

Thus MPP is T_6.

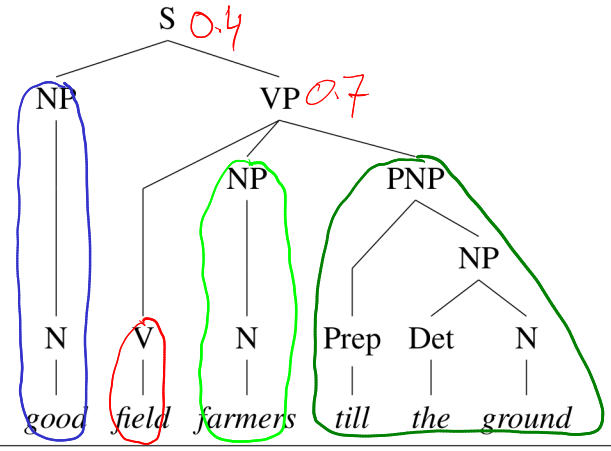
T_1:



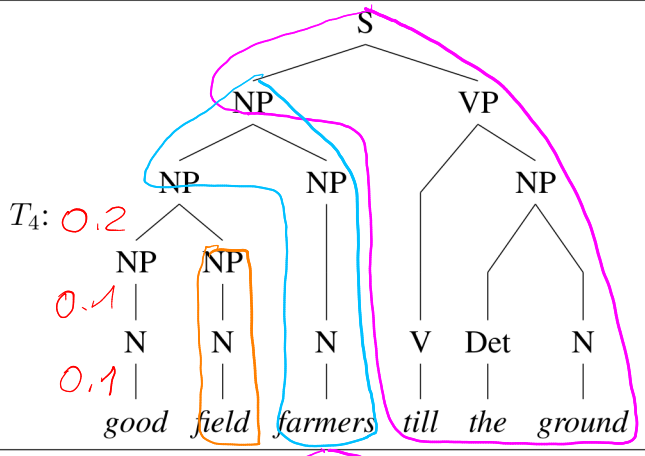
T_2:



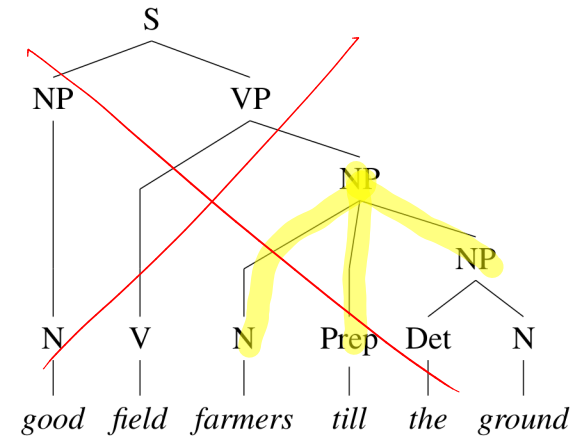
T_3:



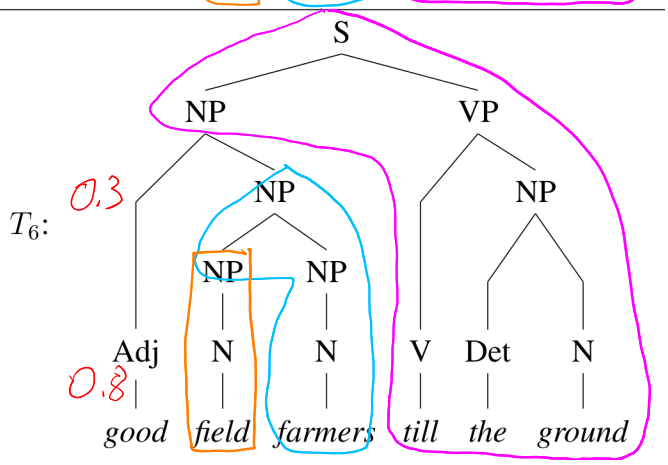
T_4: 0.2

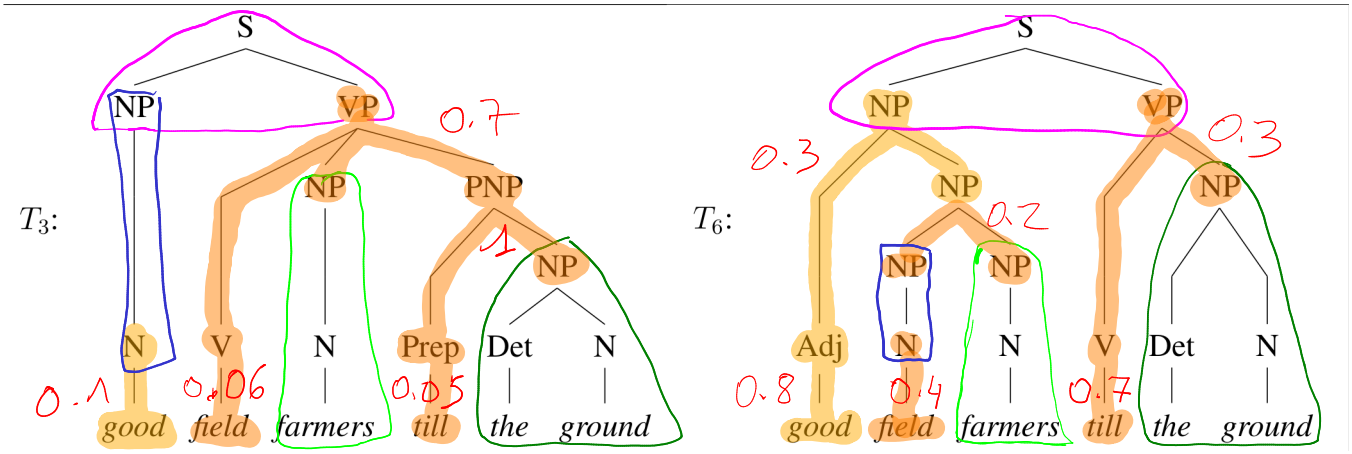


T_5:



T_6:





⑥ **[12 pt]** Fully fill in (excluding pointers) the data-structure generated by the CYK algorithm (used as an analyzer) to parse the former sentence with the above grammar:

The *first* thing to do is to transform the grammar to eCNF, for instance:

```
S  -> X1 PNP (0.6)    |  VP -> X2 PNP (0.7)
X1 -> NP VP  (1)      |  X2 -> V  NP  (1)
```

then:

S (7 = 1 + 4 + 2) NP X1					
	S NP VP (2 = 1 + 1) X1 X2				
		S NP X1			
NP (4 = 2 + 1 + 1) S X1			VP PNP X2		
NP (2)	VP NP X2			NP	
NP N Adj	NP N V	NP N	Prep V	Det	NP N Adj V
good	field	farmers	till	the	ground

The numbers here are just extra info for questions ⑦ and ⑧; there should **not**, in any case, be any duplicate of the non-terminals, otherwise the algorithm is exponential!

⑦ **[5 pt]** What is the (worst case time-)complexity of the algorithm *you* applied in question ⑥?

The key word in this question is "*you*": the question is really about how you did the parsing, it's not about the theory. Answers are either $\Theta(n^3)$ or $\Omega(\exp(n))$ depending on how you filled the chart.

Surprisingly, some (CS Master!) students do not know that algorithmic complexity is a (class of) function(s), not a number.

⑧ **[5 pt]** How many parse trees (starting with S) are there for the above sentence with the above grammar? Justify your answer.

There are 7 possible parse-trees:

- 1 with `S -> NP VP PNP` (through `X1` in cell (3,1));
- 4 with `S -> NP VP` using the 4 interpretations of `NP` in cell (3,1);
- 2 with `S -> NP VP` using the 2 interpretations of `VP` in cell (5,2).

Notice that, with the help of question ④, we know that there are at least 4 possible parse trees.

QUESTION III : Word embeddings [7 pt]

Consider a CBoW word embedding model of size 500, with a 3-token context, trained over a corpus of 2'437'832 occurrences of 53'276 different tokens, using 5 negative samples.

① **[2 pt]** How many parameters do we have to learn? Justify your answer.

$$\underbrace{2}_{\text{both } H \text{ and } M} \times \underbrace{53'276}_{\text{(shared) input(s)}} \times \underbrace{500}_{\text{hidden layer}} = 53'276'000$$

② **[5 pt]** If the corpus begins with this sentence:

Alice was beginning to get very tired of sitting by her sister on the bank

what is the contribution of the 9th token (*sitting*) to the loss function, assuming that this word is never a negative sample for any other token and that its 5 negative samples are:

Alice, *everything*, *little*, *thought*, *vanished*

Express your answer only in terms of the model parameters (and usual math functions; but no other unknown function). Introduce and explain all the notations you need.

Let's note H_{w} the projecting weights (from input to hidden layer) for word "w" and M_{w} the reverse-projecting weights (from hidden to output layer);
and let E_9 be the embedding of the context of the 9th token (*sitting*):

$$E_9 = H_{\text{very}} + H_{\text{tired}} + H_{\text{of}} + H_{\text{by}} + H_{\text{her}} + H_{\text{sister}}$$

Then, its contribution to the loss function is:

$$\begin{aligned}\log\Big(1+\exp\left(-M_{\text{sitting}}\cdot E_9\right)\Big) &+ \log\Big(1+\exp\left(+M_{\text{Alice}}\cdot E_9\right)\Big) + \log\Big(1+\exp\left(+M_{\text{everything}}\cdot E_9\right)\Big) \\ &+ \log\Big(1+\exp\left(+M_{\text{little}}\cdot E_9\right)\Big) + \log\Big(1+\exp\left(+M_{\text{thought}}\cdot E_9\right)\Big) \\ &+ \log\Big(1+\exp\left(+M_{\text{vanished}}\cdot E_9\right)\Big) \\ = f^-(\text{sitting}) &+ \sum_{w\in\mathcal{N}} f^+(w)\end{aligned}$$

with $f^-(w) = \log\Big(1+\exp\left(-M_{\text{w}}\cdot E_9\right)\Big)$, $f^+(w) = \log\Big(1+\exp\left(+M_{\text{w}}\cdot E_9\right)\Big)$
and $\mathcal{N} = \{\text{Alice}, \text{everything}, \text{little}, \text{thought}, \text{vanished}\}$

QUESTION IV : Evaluation [16 pt]

Your goal is to evaluate the performance of a fraud detection system monitoring the financial transactions carried out by an international bank for which a preliminary study has shown that the probability p_f for a transaction to be fraudulous is about 10^{-5}.

① **[1 pt]** To prepare for the evaluation, your first step is to build a reference corpus. If the transactions contained in the targeted reference corpus are drawn uniformly at random from all the transactions carried out by the bank in the past 12 months, what is the size N the targeted reference corpus should have to contain about 20 fraudulous transactions?

$N = 20/10^{-5} = 2 \cdot 10^6$ (2 millions)

② **[1 pt]** To further convert the drawn random sample of size N into a true reference corpus, each of the transactions it contains must be tagged as either "fraudulous" or "not fraudulous". The objective is to let a group of human experts perform this tagging task.
If one assumes that, on the average, an expert can process about 4 transactions per minute, and that the hourly cost of such an expert is about 300 CHF/hour, what budget B should be provisioned for the tagging task if each of the N transactions should indeed be analyzed?

First express B as a formula in N, and then provide the corresponding numerical value.

$$B = 300 \times N/240 = \frac{5N}{4} = 2.5 \cdot 10^6 \text{ CHF}$$

③ **[1 pt]** To produce the required reference corpus at a much lower cost, you decide to use the following alternative approach: you first use the bank's archive to retrieve a set of 20 transactions that have been identified as fraudulous in the past, and then you inject these transactions in the available sample of size N by replacing 20 randomly selected transactions. Finally, you perform the following approximative tagging: "fraudulous" for each of the 20 injected transactions and "non fraudulous" for all the others.

Briefly explain why the resulting tagging must indeed be considered as "approximative".

Some of the transactions tagged as not fraudulous may be fraudulous.

④ **[4 pt]** You decide to use the produced reference corpus to evaluate the available fraud monitoring system, and choose "raw accuracy" as the associated evaluation metric; within this framework:

- What would be a reasonable approximation for the score S_1 achieved by a perfect fraud detection system?
- What would be a reasonable approximation for the score S_2 achieved by a fraud detection system systematically categorizing all the transactions as "non fraudulous"?

First express S_1 and S_2 as formulas in p_f, and then provide the corresponding numerical values.

$$S_1 = 1 - p_f = 1 - 10^{-5}$$

$$S_2 \simeq 1 - p_f \simeq S_1$$

⑤ **[2 pt]** What do the results obtained in ④ tell you about the adequacy of the selected evaluation framework?

Not adequate because the proposed evaluation metric is not well suited for very unbalanced classes and because it gives about the same score to a perfect system and to a very simple baseline.

⑥ **[7 pt]** The engineer in charge of the evaluation suggests to use the following alternative approach for evaluating the available fraud detection system:
instead of asking the human experts to perform the tagging required to build a reference corpus, you first use the available fraud detection system to tag the random sample of size N, and then you ask the human experts to check the N_f transactions tagged as "fraudulous" to determine whether they are indeed fraudulous or not.

Assuming that N_f = 30, and that N_f^+ = 10 of the N_f transactions have been validated as fraudulous, compute (possibly approximately):

- the precision P and the recall R of the fraud detection system, and, for each of them, indicate whether the obtained value is or not an approximation with respect to the given data;
- the budget B_2 to be provisioned for the evaluation.

First express P, R and B_2 as formulas in the provided variables, and then provide the corresponding numerical values.

Then briefly indicate the pro's and con's of this alternative approach (**2 pt**).

$$P = \frac{N_f^+}{N_f} = \frac{10}{30} = 33\%$$

This value is exact.

$$R \simeq \frac{N_f^+}{N \cdot p_f} = \frac{10}{20} = 50\%$$

That value is approximate.

$$B2 = \frac{300}{240} N_f = \frac{300}{8} = 37.5 \text{ CHF}$$

Pro's: cheaper, faster, adequate evaluation metric

Con's: must be redone for each evaluation, recall very approximative

QUESTION V : Semantics [4 pt]

Consider the following definitions:

Definition 1 – bank *[from Middle English "banke"]*

An institution where one can place money.

Definition 2 – bank *[from French "banc"]*

A long seat with armrests and a back.

Definition 3 – stool *[from From Middle English "stol"]*

A seat for one person and without armrests.

① **[2 pt]** Use the Aristotelian principle of "Genus-Differentia" to provide a representation of the provided definitions based on semantic relations.
Try to propose a representation that uses a minimal number of relations.

```
bank_1  -(hyponym)-> institution_1
bank_2  -(hyponym)-> seat_1
bank_2  -(holonym)-> armrest_1
stool_1 -(hyponym)-> seat_1
```

It's a relation between *senses* (so don't forget the number subscripts, especially on the right part).

② **[2 pt]** Provide a synset based representation of the three above provided definitions.
Try to propose a representation that uses minimal synsets.

```
bank_1:  { bank , institution }
bank_2:  { bank , seat }
stool_1: { stool, seat }
```

QUESTION VI : Part-of-Speech tagging [15 pt]

Your goal is to build, train, and use an order-1 HMM PoS tagger. In this perspective, the design instructions you have to follow are the following:

- the tag set to use has to be the NLTK "universal" tag set (which contains 12 tags);
- all the valid surface forms are stored in an exploitable electronic lexicon containing 100'000 entries, with an average of 1.5 tags associated with each of the entries;

① **[2 pt]** How many parameters does the targeted PoS tagger need to deal with the tagging of known surface forms?
Compute the requested number of parameters without taking into account the inherent stochastic constraints they have to verify.

12 intitial probabilities;
$12 \times 12 = 144$ transition probabilities
$1.5 \times 100'000 = 150'000$ emission probabilities
$\longrightarrow 150'156$ parameters

② **[5 pt]** How do you estimate the identified parameters? In particular, explain how supervised and unsupervised methods may be used for this purpose.

- Supervised approach: using MLEs derived from an annotated corpus;
- Unsupervised approach: using Baum-Welsch algorithm on raw (i.e. non annotated corpus);
- Hybrid approach: supervised approach for computing initial values, followed by unsupervised approach for optimizing the initial values.

③ **[2 pt]** Explain why it is crucial to complement the various estimation methods with adequate smoothing mechanisms.

As a consequence of the Zipf law (specific type of power law), NL consists of many rare events which will not be observed in any given (even large) corpus ; such events will be associated with 0 probabilities if standard MLE are used, which makes them strictly impossible, and smoothing techniques must then be used to associate small but non-zero probabilities to the rare events so that they can be taken into account.

Or, more technically, the Zipfian nature of NL lead to overfitting when parameters are trained, and thus needs to be corrected by smoothing.

④ **[6 pt]** What do you propose to make you tagger able to deal with Out-of-Vocabulary forms (OoVs)? More precisely, indicate the solution you propose for each of the following OoVs categories:

(a) neologisms, such as "*EPFLish*";
(b) borrowings, such as "*rendez-vous*" (borrowed from French into English);
(c) typos, such as "*boko*" instead of "*book*".

For each a these 3 situations, indicated what you propose to do and also indicate the impact of the proposed approach on the parameters of the model.

(a) For neologisms, use morphological tools such as a (transducer based) morphological analyzer or, at least, a stemmer ;

(b) For borrowings, use open PoS tags, or, but less adequate a language identifier and an additional lexicon, or a language identifier and a translation tool ;

(c) For typos, use spelling error correction, with an FSA based edit distance with a threshold (e.g Oflazer algorithm).

All these techniques should be implemented in a « guesser », in which case (a) and (b) should not have any strong impact on the model, but (c) would require a way for dealing with possible multiple corrections.

QUESTION VII : Information Retrieval [13 pt]

① **[6 pt]** Homonymy, polysemy and synonymy are linguistic phenomena that are known to reduce the performance of standard Vector Space Information Retrieval systems.

- First provide a brief definition of homonymy, polysemy and synonymy
- and then explain why they are problematic for IR;
- illustrate your answers with relevant concrete examples.

- two words are homonymous is they have
  (a) the same spelling,
  (b) the same pronounciation,
  (c) but not the same ethymology (which entails they are not in the same lexeme),
  (d) and two different meanings ;

  example : *bat* (the wooden stick) vs. *bat* (the flying mammal) ;
- a word is polysemic if
  (a) it corresponds to a single entry in a lexicon (i.e. a single lexeme, and the same ethymology)
  (b) containing multiple related meanings ;

  example : *crown* (the headgear of a king) vs. *crown* (the top part of a tree) ;
- synonyms are
  (a) two distinct words,
  (b) that share a similar meanings ;

  example *freedom* and *liberty*.

Homonymy and polysemy are problematic for IR because they reduce the precision, as a polysemic/homonymic query can retrieve irrelevant documents ;
synonymy is problematic for IR because it reduces recall, as a query containing words that have synonyms may miss the relevant documents containing these synonyms.

② **[2 pt]** Consider a Vector Space Information Retrieval system using a simple Boolean weighting scheme (i.e. any indexing term is associated with a weight 1 in the vector representing a document if the term occurs at least once in the document, and with a weight 0 otherwise) and the cosine as proximity measure.

If "t" is an indexing term, what is the similarity $s(D, \text{"t"})$ between the query "t" and a document D associated with $|D|$ indexing terms and containing "t"?
Provide the similarity in the form of a formula in $|D|$.

$$s(D, \text{"t"}) = \frac{1}{\sqrt{|D|}}$$

③ **[3 pt]** Then assume that a "semantic tagger", i.e. a tool able to tag each of the polysemic words occurring within a document with a number identifying its meaning, is available;
and further assume that:

H1: "t" is a polysemic indexing term that can be associated with two possible meanings (i.e. the result of the tagging of "t" by the available semantic tagger is either "t/1" or "t/2");

H2: the result list retrieved for the query "t" is (D_1, D_2, D_3, D_4, D_5), with D_1 and D_3 being relevant for the meaning 1 of "t" and D_2, D_4, and D_5 being relevant for the meaning 2 of "t".

Compute the average precision AP_1("t") (resp. AP_2("t")) achieved by the IR system for the query "t" if the query is interpreted as "t/1" (resp. "t/2").

Fully justify your answers and provide the corresponding average precision values in the form of irreducible fractions.

The "ok" pattern for "t/1" is thus: 1 0 1 0 0; then

$$AP_1(\text{"t"}) = \frac{1}{2}\left(1 + \frac{2}{3}\right) = \frac{5}{6}$$

The "ok" pattern for "t/s" is: 0 1 0 1 1; then

$$AP_2(\text{"t"}) = \frac{1}{3}\left(\frac{1}{2} + \frac{2}{4} + \frac{3}{5}\right) = \frac{8}{15}$$

④ **[2 pt]** Further assume that:

H3: the available semantic tagger is integrated in the indexing pipeline as follows: the documents are semantically tagged and, in their vectorial representation, the single dimension associated with any polysemic term with k possible meanings is replaced by k dimensions, one for each of the possible meanings;

H4: if a document is relevant for the query "t/i", then it contains at least one occurrence of "t" tagged by i by the available semantic tagger;

H5: all the occurrences of a polysemic term in an given relevant document correspond to the same meaning; thus no relevant document contains both "t/1" and "t/2".

Under the assumptions H1 to H5, compute the average precisions $AvgP$("t/1") and $AvgP$("t/2") achieved by the IR system for the queries "t/1" and "t/2". Fully justify your answers and provide the corresponding average precision values in the form of irreducible fractions.

$$AvgP(\text{"t/1"}) = \frac{1}{2}(1+1) = 1$$

$$AvgP(\text{"t/2"}) = \frac{1}{3}(1+1+1) = 1$$