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)

2022 — Solution of the exam

Friday, January 28th, 2022.

QUESTION I : Language [6 pt]

① **[4 pt]** Consider the two sentences:

> *The cold has damaged one of the radiators in the building. The workers had to bring it back to the factory.*

At each of the *relevant* processing levels, indicate all the candidates the pronoun "*it*" may be referring to and what type of linguistic knowledge at that level may allow to exclude some of them.

- lexical: non relevant because anaphoric reference is a syntactic phenomenon;
- syntactic: *cold*, *one* (*of the radiators*), *building*, and maybe *factory* if forward references are considered/allowed.
  Retain only singular nouns (or noun phrases).
- semantic: *one* (*of the radiators*), *building*.
  cold is not a tangible object you can carry.
  factory: you don't brind something to itself.
- pragmatic: *one* (*of the radiators*).
  we cannot bring buildings to factories (and factories don't repair buildings)

② **[2 pt]** If a transducer is available for performing inflectional morphology, what are the input and output strings for such a transducer, when it is used as an analyzer?
Illustrate your answer with a relevant example.

input: surface form, e.g. `is`
output: canonical representation, e.g. `be+3s+IndPres`

QUESTION II : Twisted [7 pt]

Consider the following three definitions:

twist
1. A thread made from two filaments.
2. A distortion of a meaning.

deformation
1. A distortion of the shape.

① **[4 pt]** Use *semantic relations* to provide a minimal representation of these definitions that allows to distinguish them from each other.

Justify your answer, and, in particular, explain the main principle behind the approach.

```
twist.1:
twist.1  --is.a-->    thread.1    [hyponym]
twist.1  --made.of--> filament.1  [holonym]

twist.2:
twist.2       --is.a-->     distortion.1  [hyponym]
distortion.1  --apply.to--> meaning.1     [meronym]

deformation.1:
deformation.1 --is.a-->     distortion.1  [hyponym]
distortion.1  --apply.to--> shape         [meronym]

principles:
Genus (this is why there are hypernyms) Differencia (added other
relations to differenciate, e.g. deformation.1 from twist.2)
```

② **[3 pt]** Similar question as the previous one, but with synsets instead of semantic relations:

Use *synsets* to provide a minimal representation of these definitions that allows to distinguish them from each other.

```
twist:
 1. {twist, thread}
 2. {twist, distortion}

deformation:
 1. {deformation, distortion}
     [ Note that, strictly speaking, {deformation} would be enough.]
```

QUESTION III : Good news [25 pt]

You aim at developing an automatic labeling of EPFL news. For instance, you'd like the following news to be labeled "Computer Science":

```
Making quantum computers even more powerful

Engineers at EPFL have developed a method for reading several qubits
- the smallest unit of quantum data - at the same time.
Their method paves the way to a new generation of even more powerful
quantum computers.
```

① **[3 pt]** What kind of NLP task is it? What specific method(s) do you propose for this task (give at most two)? Briefly justify your answer.

This is text classification (supervised) or text clustering (unsupervised). In this case, it's most probably classification so as to have categories that make sense, but it is unclear whether we have such a corpus (supervision) or not.

Methods:

- supervised: KNN, Naive Bayes, neural (MLP, SVM, RNN+FF, transformers), ...
- unsupervised: k-means, dendrograms, ...

Many assumed its classification without dicussing the origin of the classes nor the need for supervised data.

② **[6 pt]** What techniques and processing steps do you foresee to fulfill this task (end-to-end)? For each step, provide both a description and, as an illustrative example, an explanation of how the first sentence of the above example news:

```
Engineers at EPFL have developed a method for reading
several qubits - the smallest unit of quantum data -
at the same time.
```

would be processed (you can directly annotate on it, when that's easier).

We should consider the usual text processing steps:

- Tokenization (split into words using separators, maybe removing punctuation (unclear)).

  `Engineers|at|EPFL|have|developed|a|method|for|reading`
  `several|qubits|the|smallest|unit|of|quantum|data|at|the|same|time`

- That step may also include some kind of normalization, e.g. lowering capital letter at beg. of sentence;
  for instance:

  `engineers|at|EPFL|have|developed|a|method|for|reading`
  `several|qubits|the|smallest|unit|of|quantum|data|at|the|same|time`

- It may also include (N)ER (e.g. `EPFL` here) or some way to deal with Out-of-Vocabulary forms.
  for instance:

```
engineers|at|<NE>EPFL</NE>|have|developed|a|method|for|reading
several|qubits|the|smallest|unit|of|quantum|data|at|the|same|time
```

- Part-of-Speech tagging (could even come before NERs and Oov if good guesser)
  for instance:

  ```
  engineers/N|at/Prep|<NE>EPFL</NE>/N|have/V|developed/V|a/Det|
  method/N|for/Prep|reading/V|several/Adj|qubits/N|the/Det|
  smallest/Adj|unit/N|of/Prep|quantum/N|data/N|at/Prep|the/Det|
  same/Adj|time/N
  ```

- lemmatization (or stemming), to reduce lexical variability;
  for instance:

  ```
  engineer/N|at/Prep|<NE>EPFL</NE>/N|have/V|develop/V|a/Det|
  method/N|for/Prep|read/V|several/Adj|qubit/N|the/Det|
  small/Adj|unit/N|of/Prep|quantum/N|data/N|at/Prep|the/Det|
  same/Adj|time/N
  ```

- filtering, based on frequencies, stop-words, PoS tags, to remove entities which carry less meaning for the target application;
  for instance: get rid of `the/det`, `of/Prep`, etc.; e.g.:

  ```
  engineer/N|<NE>EPFL</NE>/N|develop/V|method/N|read/V|
  qubit/N|small/Adj|unit/N|quantum/N|data/N|same/Adj|time/N
  ```

  depending on design, this step might come before lemmatization;
- then finally proper representation for the chosen technique: bag-of-words (counting), or parameters look-up (probabilities), or word embeddings.
  for instance: same as above with "`(, 1)`" around each.

Lack of examples *for each step*. Forget to deal with punctuation or OoV.

Some students seems to think they can do PoS tagging *after* stop-words removal.

③ **[4 pt]** To simplify, we consider only two possible labels: `LS` (Life science) and `CS` (Computer Science); and limit the processing to the news title only.

What would be the probability for the following title:

```
Improved motor, sensory, and cognitive recovery after stroke.
```

to be labeled with each of the two above labels, using a (1-gram) Naive Bayes system, trained on a corpus where:

- 45% of the news are `LS` (the rest being `CS`, no news in common);
- the relative frequency of each of the two labels for each of the following words are:

	LS	CS
improved	0.49	0.51
motor	0.80	0.20
sensory	0.60	0.40
cognitive	0.75	0.25
recovery	0.42	0.58
stroke	0.65	0.35

- the relative frequency of each of the following words for each of the two labels are:

	LS	CS
improved	$7 \cdot 10^{-6}$	$6 \cdot 10^{-6}$
motor	$5.4 \cdot 10^{-6}$	$1.1 \cdot 10^{-6}$
sensory	$5.5 \cdot 10^{-6}$	$3 \cdot 10^{-6}$
cognitive	$14.7 \cdot 10^{-6}$	$4 \cdot 10^{-6}$
recovery	$4.4 \cdot 10^{-6}$	$5 \cdot 10^{-6}$
stroke	$4.5 \cdot 10^{-6}$	$2 \cdot 10^{-6}$

Provide your answer as a product of numbers (no need for a final value) and *fully justify* your answer (explaining assumptions where needed).

For label L:

$$P(\texttt{L}|\texttt{words}) \propto P(\texttt{L}) \times \prod_{\texttt{words}} P(\texttt{word}|\texttt{L})$$

Too many write an equalitiy here, rather than "proportional to".

For label LS: $\propto 0.45 \times 7 \times 5.4 \times 5.5 \times 14.7 \times 4.4 \times 4.5$

For label CS: $\propto 0.55 \times 6 \times 1.1 \times 3 \times 4 \times 5 \times 2$

(with the same proportional coefficient)

④ **[2 pt]** What is the fundamental difference between Naive-Bayes classification and multinomial logistic regression?

Naive-Bayes classification is a *generative* model (focusses on joint probability), whereas multinomial logistic regression is a *discriminative* model (focusses on posterior probability).

Naive-Bayes assumes conditionnal independence of indexing terms knowing the class. Multinomial logistic regression directly models the posterior probability with features (weights) relating indexing term and class, which are learned during training.

Notice that multinomial logistic regression, by expressing the posterior probability as a product, is also making a simplification assumption similar to probabilistic conditionnal independence made for Naive Bayes (although here it's not a product of probabilities).

⑤ **[5 pt]** In order to improve your process (especially for OoV tokens), you consider using a 4-gram character model estimated on a corpus containing 1'000'000 occurrences of 4-grams.

Parameters are estimated using additive smoothing with a Dirichlet prior with parameters uniformly set to some value α.

Knowing that non-occurring 4-grams have parameters estimated to 10^{-7} and hapaxes[1] have parameters estimated to $1.05 \cdot 10^{-6}$, what is the estimated value of parameters, the maximum-likelihood estimation of which would have been 10^{-5}?

Express your answer as a numerical value in the form $a \cdot 10^{-b}$ and *fully justify* your answer.

The estimated value for a 4-gram occurring n times is

$$\frac{n+\alpha}{N+\alpha \times T}$$

where N is the number of occurrences (1'000'000) and T the number of possible 4-grams ($|A|^4$ with $|A|$ the size of the alphabet).

Let p_0 and p_1 respectively be the estimates for non-occurring 4-grams and for hapaxes. We thus have

$$p_0 = \frac{\alpha}{N+\alpha \times T} \quad \text{and} \quad p_1 = \frac{1+\alpha}{N+\alpha \times T}$$

And thus, the estimated value for a 4-gram occurring n times is

$$\frac{n+\alpha}{N+\alpha \times T} = n \times (p_1 - p_0) + p_0 = n \times p_1 - (n-1) \times p_0$$

[1] 4-grams occurring only once.

If the maximum-likelihood estimation of a parameter is 10^{-5}, then $n = 10^{-5} \times N = 10^{-5} \times 10^6 = 10$, and thus its estimated value is

$$10 \times 1.05 \cdot 10^{-6} - 9 \times 10^{-7} = 10.5 \cdot 10^{-6} - 0.9 \times 10^{-6} = 9.6 \cdot 10^{-6}$$

A striking number of students made the following mistake:

$$a + x = b \implies x = \frac{b}{a} \qquad (!!)$$

You finally decide to use transformers to achieve the task.

⑥ **[3 pt]** What are the advantages of transformers over RNN?

(a) No need to model text sequences one word at a time;
able to make any useful, especially long-term, mixture (*attention* mechanism), avoiding the vanishing gradiant problem;
able to make some of the past more relevant to the future.
In RNN or vanilla LSTM, it's challenging to learn *any* long-range dependencies (no DIRECT way to model it).

(b) In recurrent models: can't parallelize recurrent computations (time-step sequential dependencies).
In transformers: "states" computation can be done in parallel (*self*-attention mechanism), so no longer propagating states forward in time.

(c) (*multi-head* attention) combine **sub**parts (subspaces/projection) of the input vectors/"states" rather than the full vector

Nobody mentionned the multi-head aspect (I thus removed it from the grading scale).

⑦ **[2 pt]** Why and how to encode the positions of words in transformers?

- **Why:** we've lost word order information (self-attention computed in parallel) –> kind of BoW representation,
  so we shall encode it somehow (since word order still seems crutial for NL).
- **How:** add an additional embedding to the input word representation that represents a position in the sequence $\longrightarrow$ "Position Embedding".
  Learn it from scratch or represent it as some kind of wave function (like e.g. a phase of a sine)

QUESTION IV : Parsing [31 pt]

Consider the following SCFG and lexicon:

```
S   -> NP VP PNP   [0.4]
S   -> NP VP       [p1]
NG  -> N           [0.2]
NG  -> N N         [p2]
NG  -> AG N        [0.3]
AG  -> Adj         [p3]
AG  -> AG Adj      [0.2]
NP  -> Det NG      [0.5]
NP  -> NP PNP      [p4]
VP  -> V PNP       [0.25]
VP  -> V           [0.15]
VP  -> V NP PNP    [p5]
VP  -> V NP        [0.25]
PNP -> Prep NP     [p6]
```

```
garden:N
garden:V
gossips:N
gossips:V
in:Prep
lady:N
little:Adj
little:Adv
rabbit:N
the:Det
white:Adj
white:N
with:Prep
young:Adj
young:N
```

where `p1` to `p6` are (non-null) numbers between 0 and 1; and consider the following sentence:
`the little white rabbit gossips with the young lady in the garden`

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

32: $2(\textit{little}) \times 2(\textit{white}) \times 2(\textit{gossips}) \times 2(\textit{young}) \times 2(\textit{garden})$; all the other words being unambiguous.

② **[1.5 pt]** Propose some values for `p1` to `p6` for the grammar to be a correct SCFG.

$$p_1 = 0.6, \quad p_2 = 0.5, \quad p_3 = 0.8, \quad p_4 = 0.5, \quad p_5 = 0.35, \quad p_6 = 1$$

You make use of some CYK algorithm implementation to parse the above sentence with the above grammar and get the data structure displayed on the next page.
Notice that not all the details are provided. Only a *subset* of this data structure is represented. When a pair of pointers is displayed, also are all the other pairs of pointers for the same non-terminal. Useful missing information can be reconstructed unambiguously where needed (it's up to you to do so).

③ **[1.5 pt]** Parser internal data structures usually represent three kind of information:

(a) "what has been done up to here";

(b) where "this" starts ("this" refers to former "what has been done up to here");

(c) and where "this" ends.

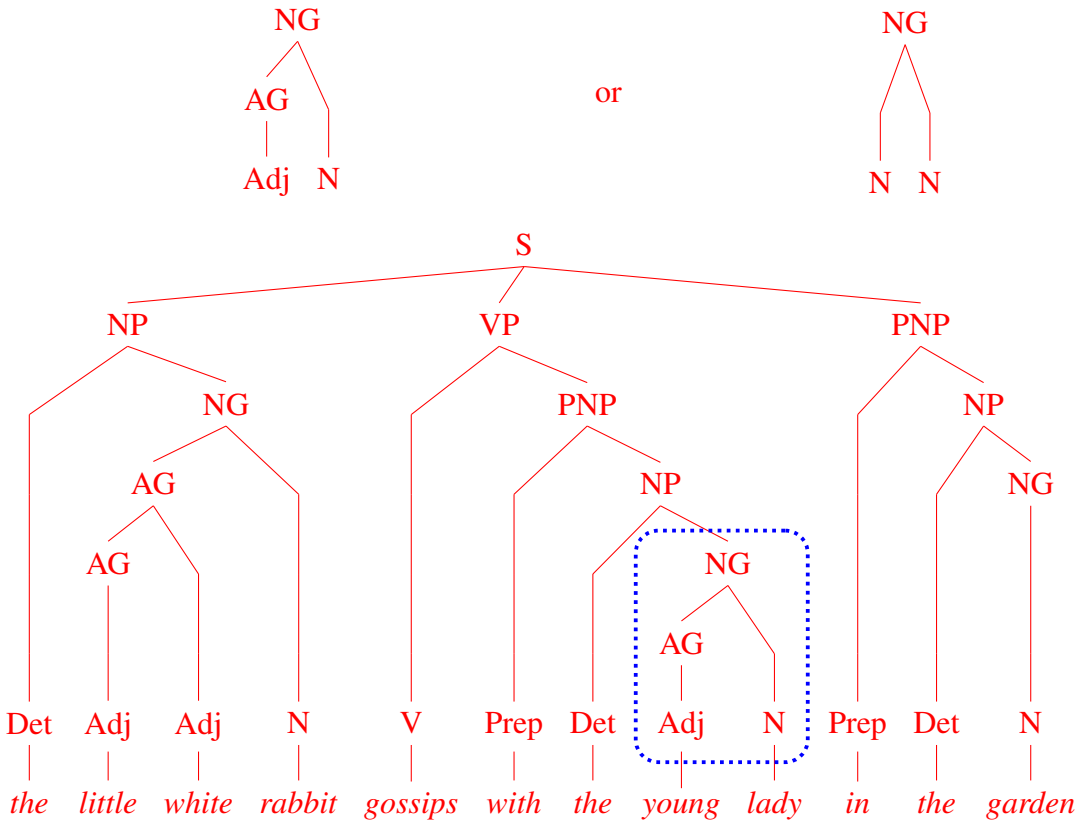
In a CYK parser, how are the above three pieces of information represented?

- "what has been done up to here": cell content, typically non-terminals;
- where "this" starts: column number (j);
- and where "this" ends: "diagonal number", more precisely: $i + j - 1$, where i is the row number.

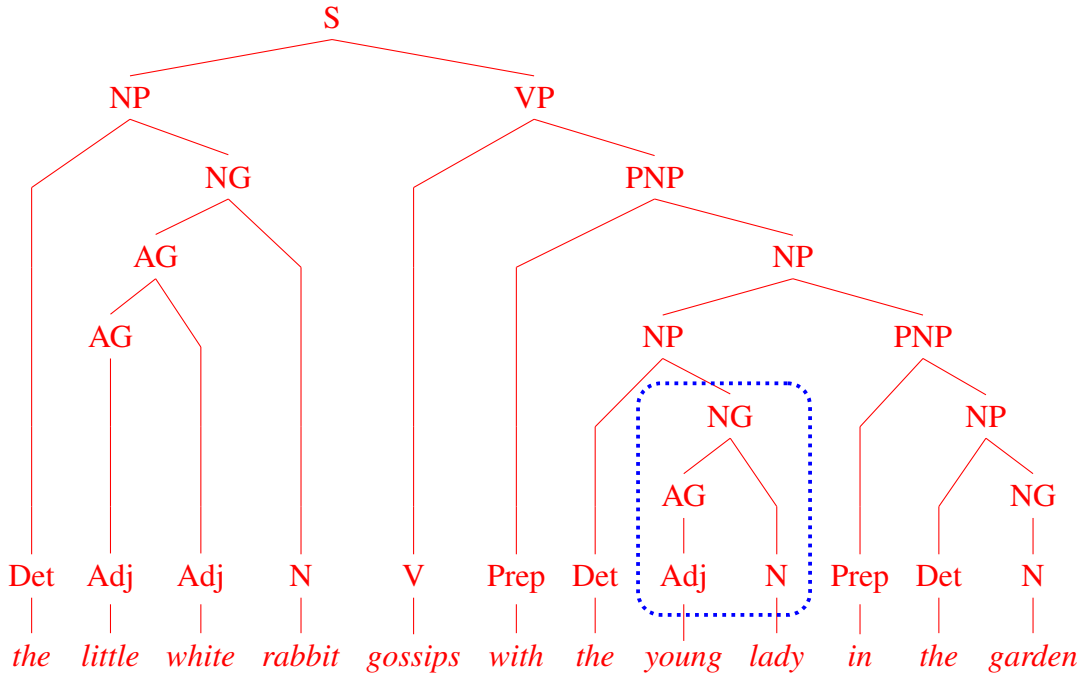
④ **[3.5 pt]** Completely fill (*without* pointers) the first row of the data structure provided on the next page.

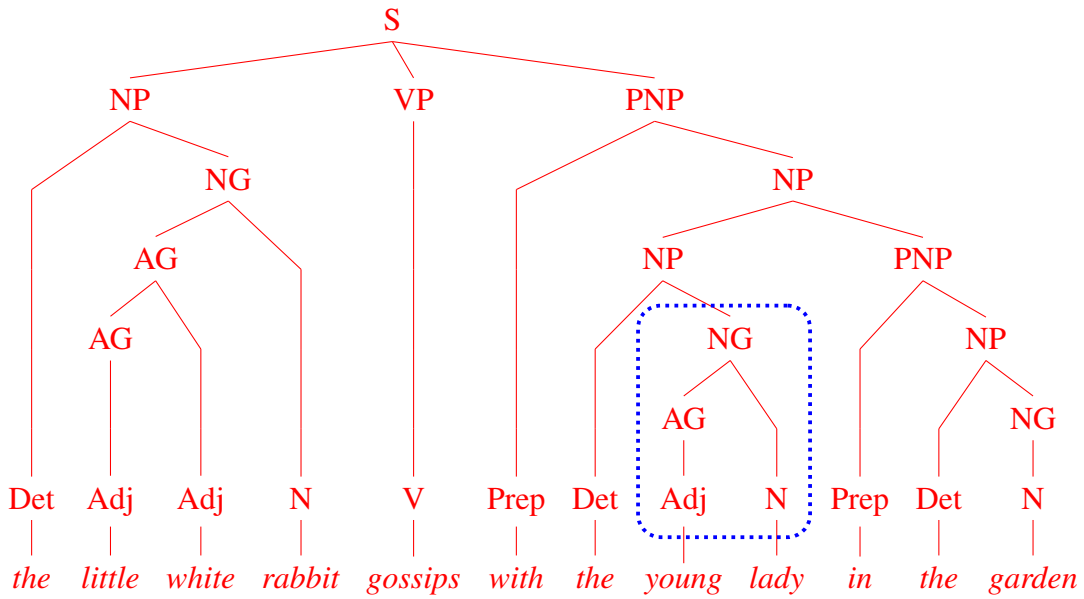
⑤ **[5 pt]** Draw one of the parse trees of the input sentence (reported below):

Any of the following solutions is valid, where the blue-surounded subtree can either be:



continues on back ☞





⑥ **[2.5 pt]** What does represent the `X` in the first column fifth row (of the chart provided on the next page; numbered from bottom)? Where does it come from? Fully justify your answer.

This is the extra non-terminal coming from the CNF transformation of the rule `S -> NP VP PNP` into `S -> X PNP` and `X -> NP VP`.

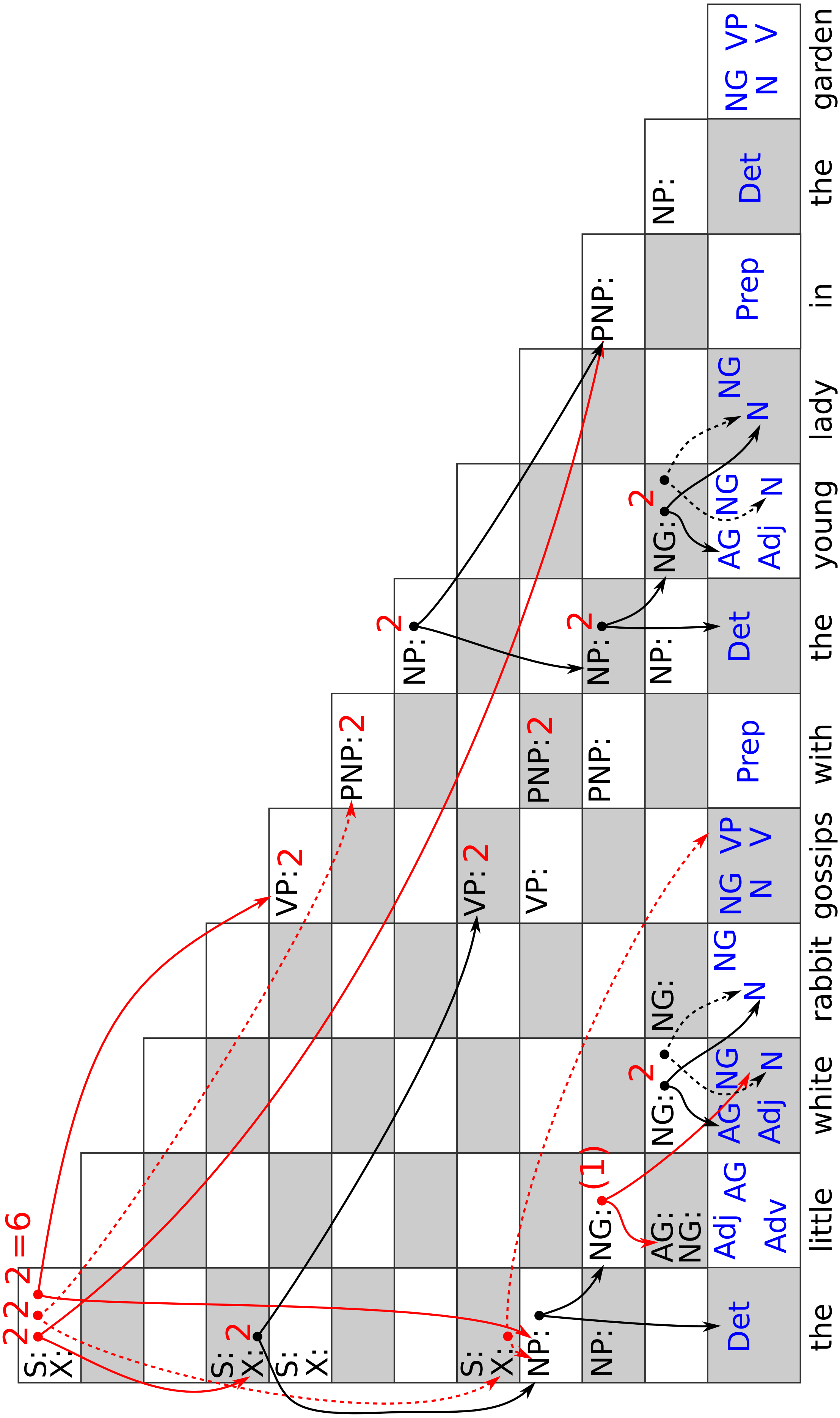
In this very case (fifth row), `NP` covers "*the little white rabbit*" and `VP` covers "*gossips*".

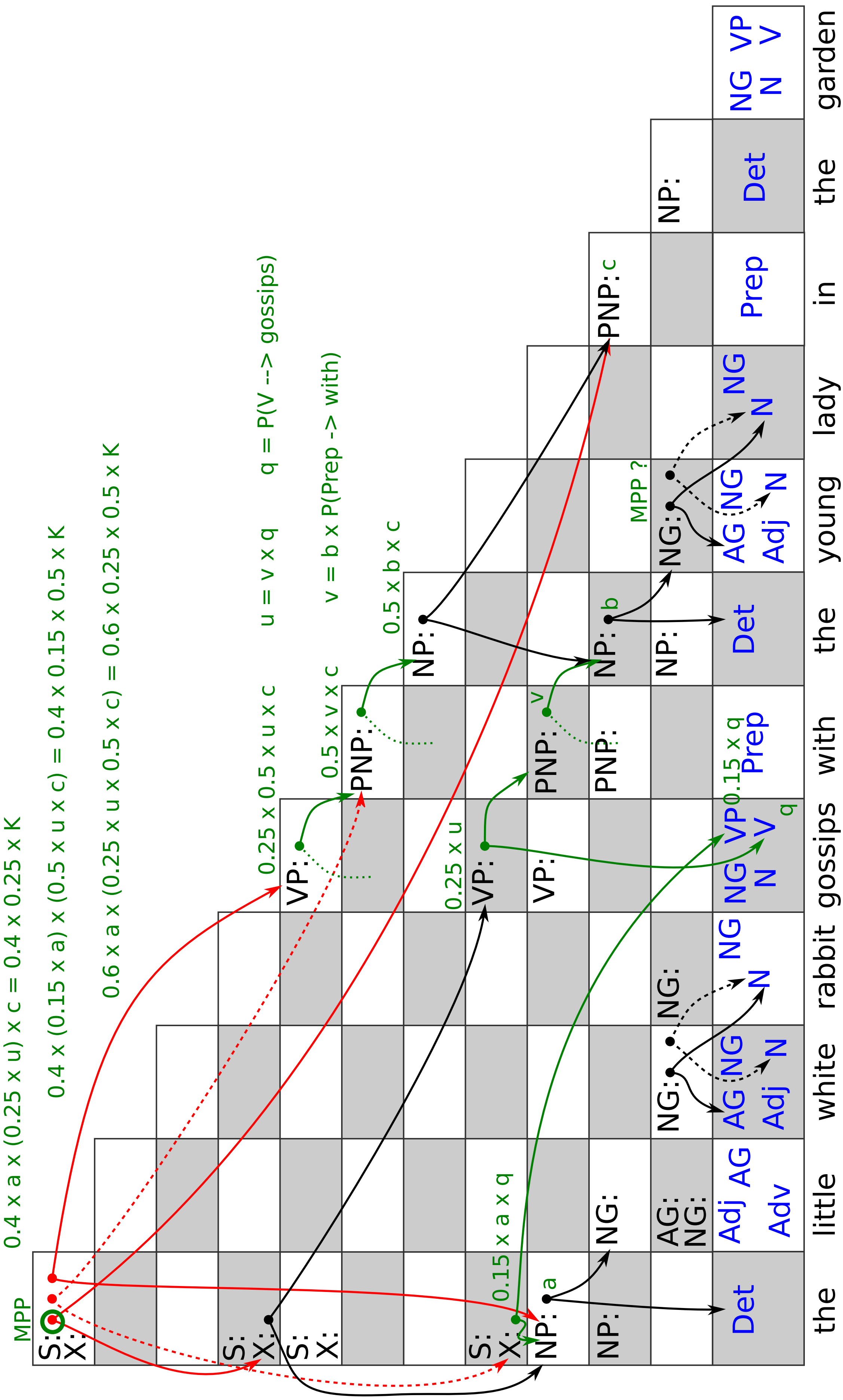
⑦ **[9 pt]** How many parse trees does the input sentence have? Answer this question by *partially* annotating the chart on the next page and reporting a clear answer below. We do not ask you to fully fill the chart: feel free to only add pieces of information that are relevant for you to answer.

The easiest way is to simply mark on the chart, bottom-up, the number of subtrees which are greater than 1 (see next page).

There are 3 possible starts for the trees (two with `S -> NP VP PNP` and one with `S -> NP VP`), for each of which we have to sum up the number of subtrees. The number of subtrees is the product of the number of subtrees on the left times the number on the right: in this case, it's easy because in all cases it's either 1×1 or 1×2, the latest occuring only when "*young lady*" is involved.

The total final value is thus $2 \times 1 + 1 \times 2 + 1 \times 2 = 6$.

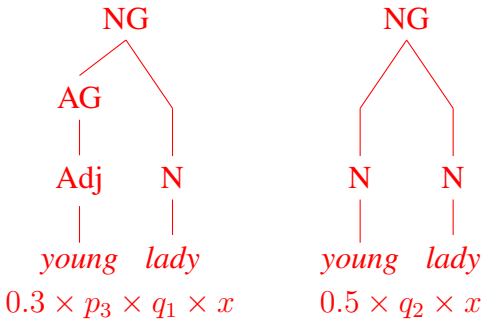

the
little
white
rabbit
gossips
with
the
young
lady
in
the
garden
2 2 2 =6


0.4 x a x (0.25 x u) x c = 0.4 x 0.25 x K
0.4 x (0.15 x a) x (0.5 x u x c) = 0.4 x 0.15 x 0.5 x K
0.6 x a x (0.25 x u x 0.5 x c) = 0.6 x 0.25 x 0.5 x K
u = v x q
q = P(V --> gossips)
v = b x P(Prep -> with)
MPP
MPP ?
the little white rabbit gossips with the young lady in the garden

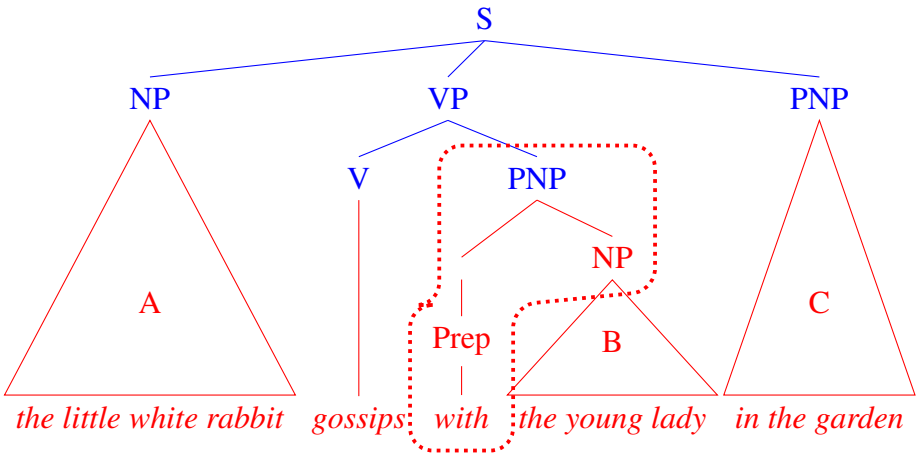
⑧ **[6 pt]** Draw the most probable parse tree for the input sentence. Justify why it is the most probable one (if it helps, you could also annotate the chart on one of the two previous pages).

The six parse trees are made of three different start (top rule) combined with *the same* possible two choices for "*young lady*". The MPP is then the most probable "start" (with all its implications: the most probable non common part among the three groups of two trees) combined with the most probale parse for "*young lady*".

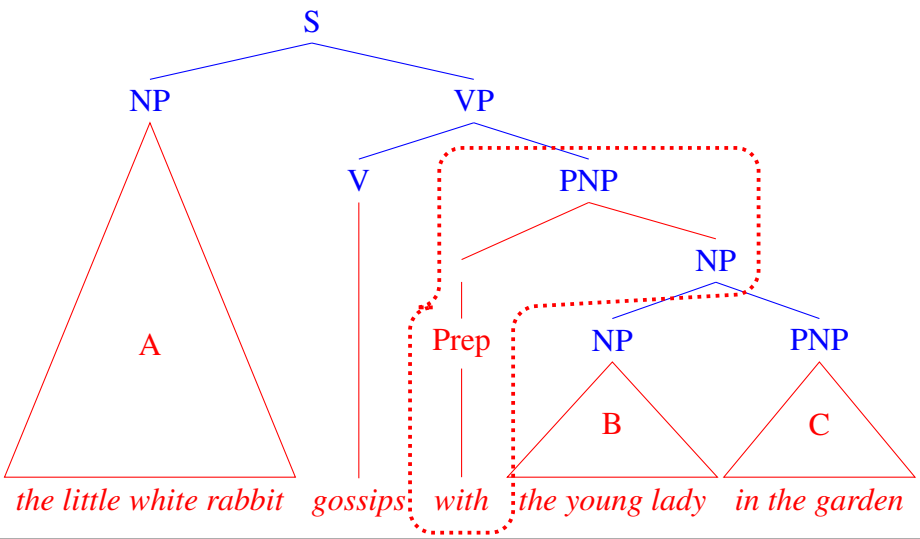
For this second choice, let's note $q_1 = P(\texttt{Adj -> young})$ and $q_1 = P(\texttt{N -> young})$. The MPP depends whether $0.3 \times p_3 \times q_1$ is greater than $0.5 \times q_2$ or not (i.e. how $0.24\, q_1$ compares to $0.5\, q_2$).



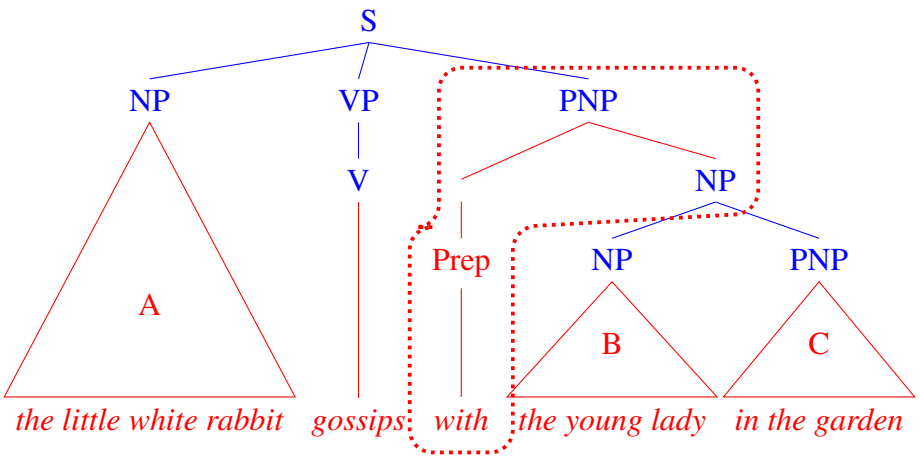
Regarding the first choise (which of the three groups implied by the top rule), the easiest thing to do in such a simple case (3 choices), is maybe to do it directly on the trees, factorizing the common factors (denoted by K below):



$0.4 \times 0.25 \times K$
$= 0.1\, K$



$p_1 \times 0.25 \times p_4 \times K$
$= 0.6 \times 0.25 \times 0.5 \times K$
$= 0.075\, K$



$$0.4 \times 0.15 \times p_4 \times K$$
$$= 0.4 \times 0.15 \times 0.5 \times K$$
$$= 0.03\,K$$

The most probable parse is then the first one, with as tree B, the most probable of the two trees for "*the young lady*".

Another (more usual) way to find the most probable of the three choices for the top, is to annotate the CYK chart with the MPP in each cell; again factorizing (= not expliciting) the common factors, as done on the chart two pages before.

For the choice of the top level, several students thinks that choosing the most probable first rule leads to the MPP; which is wrong, the whole product of all implied subtrees had to be considered.

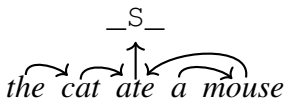
For the choice of for "*the young lady*", many forgot the lexical rules.

QUESTION V : To parse or to tag? [10 pt]

The simplest form of dependency-based grammars can be defined as follows (example below):

> For any sequence of words $S = w_1 w_2 \cdots w_n$, each of the n words but one is associated to another word in the sentence, called its "*head*" (while the original word is called a "*dependency*").
> The unique word not associated with a head is having a special head conventionally denoted by "_S_" (representing the root of the sentence).
>
> By definition, the head of a dependency (i.e. the word it is associated with) is the word in the sentence with which the dependency is most strongly connected from a syntactic perspective.
> As the verb is most often the most important word in a sentence (from a syntactic perspective), it is usually the one not associated to any other word.

For example, in the sentence "*the cat ate a mouse*", we have:



- the determiner "*the*" has "*cat*" as head because it is strongly (syntactically) connected with the noun it is characterizing;
- similarly, the determiner "*a*" has "*mouse*" as head;
- the subject "*the cat*" is strongly connected to the verb "*ate*", and therefore, "*cat*" (as the head of the subject) has "*ate*" as head;
- similarly, the direct object "*a mouse*" is also most strongly connected to the verb "*ate*", and, its head "*mouse*" thus also has "*ate*" as head;
- finally, "*ate*" is the unique word not associated with a head, and is thus having the special head "_S_".

The above information can be represented as a sequence of (dependency, head) pairs. For convenience, the dependent word will be written in lowercase, and the head word in uppercase. For the above example sentence, the sequence is thus:

(the, CAT) (cat, ATE) (ate, _S_) (a, MOUSE) (mouse, ATE)

This sequence can also be interpreted as a tagging of the sequence "*the cat ate a mouse*" with a very specific tag set consisting of the special head "_S_" and the words of the sentence themselves as heads (thus in upper case).

① **[5 pt]**

(a) If the above tagging is implemented as an order-1 HMM, which are the different parameters of the model?

(b) If a lexicon containing L words is used, how many (not necessarily free) parameters does it corresponds to?

(c) When comparing to an order-1 HMM **Part-of-Speech** tagger operating with L words and T tags, what can you say about the estimation of the parameters of the proposed model?

Justify your answers.

(a)

- Initial probabilities of the form P(TAG);
- Transition probabilities of the form P(TAG 2| TAG1);
- Emission probabilities of the form P(word | TAG);

(b)

$$(L+1) + (L+1)^2 + L \times (L+1) = 2 \times (L+1)^2$$

(c) Compared to order-1 HMM PoS tagger: $L \times T + T^2 + T \simeq L \times T \ll L^2$: new model is much harder to estimate.

② **[2 pt]** Would you a priori force some of these parameters to be zero (or simply remove them)? If yes, which parameters and why? Justify your answer.

$P(\texttt{\_S\_}|\texttt{\_S\_})$ shall be 0 because only one word can be tagged by _S_.

Some probabilities with few linguistic plausibility (e.g. $P(V|Det)$) may also be 0.

③ **[3 pt]** If the most probable tagging produced for a sentence by the proposed model contains a sequence of identical tags, for instance:

the/CAT dirty/CAT black/CAT cat/ATE ate/_S_a/MOUSE nice/MOUSE white/MOUSE mouse/ATE,

what can be said about the most probable tagging of the same sentence, but where the words associated with the sequence of identical tags have been arbitrarily permuted?
Fully justify your answer.

It will be the same since the same formula is maximized (multiplication is commutative).

QUESTION VI : Automated translation of technical manuals [14 pt]

The goal is to set up an evaluation procedure for automated translation systems for technical manuals. In this perspective, a corpus of English documents has been collected. In order to produce the corresponding French translations, each of the available documents has been submitted to the automated translation system.

① **[2 pt]** What are the main characteristics of the collected corpus of English documents that should be taken into account for the evaluation made with these documents to be credible? Justify your answer.

Large enough and representative enough.

The decision has been taken to implement the evaluation procedure in the form of n-ary classification, first at the whole document level, and then at the sentence level.

② **[3 pt]** Indicate at least two possible values for n. For each of these values n, propose a corresponding set of classes and explain what each of these classes should correspond to.

$n = 2$: good or bad (good translation vs. bad translation)

$n > 2$: finer grain: e.g. $n = 3$: Good vs. Average vs. Bad or Good vs. Neutral vs. Bad;

③ **[3 pt]** For each of the classifications proposed in the previous question, indicate what are its main advantages and drawbacks.

Higher $n \longrightarrow$ more subjective classification task $\longrightarrow$ lower inter-annotator agreement.

Higher $n \longrightarrow$ more precise classification task.

Lower $n \longrightarrow$ easier classification task.

④ **[2 pt]** For the evaluation at the sentence level, indicate the main problems that should be anticipated, if the evaluation has to be implemented in the form of a n-ary classification of (English sentence, French sentence) pairs.

- How to identify sentences?
- How to align source sentences with corresponding target sentences?
- Context? Lack of context when at sentence level.

⑤ **[4 pt]** Assume that to perform your evaluation you asked two experts to annotate a corpus made of 10000 items with 2 classes, either A or B.
The first expert annotated 7500 items as A; the second expert annotated 3000 items as B; and 6500 items were rated as A by both experts.

What is your opinion about using this corpus for you evaluation? Fully justify your answer.

	expert 1		
	A	B	
A	6500	500	7000
B	1000	2000	3000
	7500	2500	(8500)

raw agreement: 0.85

chance agreement: $0.7 \times 0.75 + 0.3 \times 0.25 = \frac{21}{40} + \frac{3}{40} = \frac{24}{40} = 0.6$

$$\kappa = \frac{0.85 - 0.6}{0.4} = \frac{2.5}{4} = \frac{10}{16} = 0.625$$

Although positive (better than chance), this κ value is too low to make use of that corpus: the experts don't agree enough one with each other to trust their decisions.

QUESTION VII : IR for CV processing [14 pt]

A large human-resources (HR) consulting company is planning to implement an automated system for processing the hundreds of resumes they have to monitor every month.

Currently, the resumes are first extracted from various internet sources by a web retrieval search engine, and then post-processed manually. The manual post-processing consists in extracting from each resume interesting "information bits" (pieces of the resume relevant for a field), and then storing them in a pre-defined template consisting of several fields (name, surname, age/date-of-birth, nationality, current position, previous positions, education, domains of expertise, hobbies, etc.).

The goal of the new automated module is to mimic, as efficiently as possible, the existing manual post-processing, and the HR company is planning that both processing pipelines (manual and automated) will co-exist, at least for a certain period of time, with a fraction of the staff involved in the post-processing evaluating the outputs of the automated module with a binary bad/good annotation at the field level.

① **[4 pt]** The first step of the design of the new automated system is to decide whether the existing web retrieval system is adequately tuned to feed the targeted automated resume processing module.

(a) If a web retrieval system is mainly used to generate input for some manual processing (as the company is currently doing it), shall it be tuned to favor precision, or to favor recall?

(b) If the goal is to use it to feed an automated processing module, should this tuning be modified?

Justify your answers.

(a) Should favor precision; To reduce the amount of manual post-processing;

(b) Should favor recall if good enough performance; To be more exhaustive

To speed up the design of the targeted automated module, the decision has been taken to re-use as much as possible the information retrieval technology used in the existing web retrieval engine. More precisely, each of the resumes to post-process is first automatically split into "information bits", and the filling of the fields present in the pre-defined template is then implemented as an information retrieval task performed, for each of the fields, on the collection of "information bits" generated for the resume.

② **[3 pt]** Challenge the decision to systematically use the information retrieval approach for all the fields in the predefined template.
Justify your answer and provide some illustrative examples.

Only for the fields with textual content; possible bad performance on others;
Because more structured fields (dates, age, etc.) or fields with few linguistic content (names, etc,) may be better processed with other techniques.

③ **[3 pt]** When information retrieval is used for a given field, how should the corresponding query/queries be generated?

The past manual pre-processing should be exploited: query = concatenation of past values

④ **[2 pt]** How would you suggest to deal with the fields, such as "Education", that may have to contain several "information bits"?

How: duplicate the field; or make subfields; or put multiple values in the field;

Use relevance to define filling rules to select where each information bit should be inserted.

⑤ **[2 pt]** Assume that for the field "Education" of a given resume in the evaluation corpus, the system retrieved the following "information bits", which have been evaluated as relevant (or not) by some expert:

information bit ID	expert annotation
`#1e03fd`	relevant
`#a7974e`	–
`#b3e17f`	relevant
`#fca981`	relevant
`#e6511f`	–

Assuming that for this field of this resume, the expert found overall 7 relevant "information bits", what are the precision and the recall of the system for this field of the considered resume? Briefly justify your answer.

$P = \frac{3}{5}$; $R = \frac{3}{7}$