01 **Text classification** **(3 points)**

a) The evaluation of a text classifier produced the following confusion matrix. The marked cell gives the number of times the system classified a document as class `C` whereas the gold-standard class for the document was `A`.

	A	B	C
A	58	6	1
B	5	11	2
C	0	7	43

Set up fractions for the classifier's precision with respect to class `C` and its recall with respect to class `A`. (You do not have to simplify the fractions.)

b) A certain Naive Bayes classifier has a vocabulary consisting of 48,359 unique words. Suppose that training the classifier on a collection of movie reviews gave

$$\#(\text{shrek}, \text{pos}) = 45\,, \quad \#(\text{shrek}, \text{neg}) = 0\,, \quad N_{\text{pos}} = 712480\,, \quad N_{\text{neg}} = 636767\,,$$

where $\#(w, c)$ denotes the number of occurrences of w in documents with class c and N_c denotes the total number of word occurrences in documents with class c. Estimate $P(\text{shrek} \mid \text{pos})$ and $P(\text{shrek} \mid \text{neg})$ using Maximum Likelihood estimation with add-one smoothing. Answer with fractions. (You do not have to simplify the fractions.)

c) Practical implementations of a Naive Bayes classifier often use log probabilities. Sketch the graph for the function $f(x) = \log x$ within the interval $0 \leq x \leq 1$. What is the function's value for $x = 1$?

Sample answers:

a) Fractions for precision and recall:

$$\text{precision with respect to class } \texttt{C} = \frac{43}{1+2+43} = \frac{43}{46}$$
$$\text{recall with respect to class } \texttt{A} = \frac{58}{58+6+1} = \frac{58}{65}$$

b) Estimated probabilities:

$$P(\text{shrek} \mid \text{pos}) = \frac{\#(\text{shrek}, \text{pos}) + 1}{N_{\text{pos}} + 1 \cdot |V|} = \frac{45+1}{712480 + 1 \cdot 48359}$$
$$P(\text{shrek} \mid \text{neg}) = \frac{\#(\text{shrek}, \text{neg}) + 1}{N_{\text{neg}} + 1 \cdot |V|} = \frac{0+1}{636767 + 1 \cdot 48359}$$

c) For the graph, see the slides for Lecture 1, slide 50. The important property is that $f(1) = 0$ and that the values of f approach negative infinity ($-\infty$) as x approaches 0. The function's value for $x = 1$ is $f(1) = 0$.

02 **Language modelling** **(3 points)**

The Corpus of Contemporary American English (COCA) is the largest freely-available corpus of English, containing approximately 560 million tokens and 1.5 million unique words. We have the following counts of unigrams and bigrams: *white*, 256,091; *snow*, 38,186; *purple*, 11,218; *white snow*, 122; *purple snow*, 0.

a) Estimate the probabilities $P(\textit{white})$ and $P(\textit{snow} \mid \textit{white})$ using maximum likelihood estimation. Use the fact that $\#(u\bullet) = \#(u)$. Answer with fractions containing concrete numbers.

b) Estimate the bigram probability $P(\textit{snow} \mid \textit{purple})$ using maximum likelihood estimation with add-k smoothing, $k = 0.01$. Answer with a fraction containing concrete numbers.

c) We train two trigram models on the COCA corpus using maximum likelihood estimation: one without smoothing, and one with add-one smoothing. We compute the entropy of both models on the training data. What can you say about the two entropy values? Provide an informal explanation.

Sample answers:

a) Maximum likelihood estimation:

$$P(white) = \frac{256{,}091}{560{,}000{,}000} \qquad P(snow \mid white) = \frac{122}{256{,}091}$$

b) Maximum likelihood estimation with add-k smoothing, $k = 0.01$:

$$P(snow \mid purple) = \frac{0 + 0.01}{11{,}218 + 0.01 \cdot 1{,}500{,}000}$$

c) The smoothed model has a higher entropy than the unsmoothed model. Add-one smoothing *increases* the total probability of trigrams that do*not* occur in the training data but *decreases* the total probability of the trigrams that *do* occur. As a consequence, the entropy of the training data goes up.

03 Part-of-speech tagging (3 points)

a) The evaluation of a part-of-speech tagger produced the following confusion matrix. The marked cell gives the number of times the system tagged a word as a verb (VB) whereas the gold standard specified it as a noun (NN).

	NN	JJ	VB
NN	58	6	1
JJ	5	11	2
VB	0	7	43

Set up fractions containing concrete numbers for the tagger's (i) recall on verbs and (ii) precision on nouns. You do not have to simplify the fractions.

b) The following matrices specify (parts of) a hidden Markov model. The marked cell specifies the probability for the transition from BOS to AB.

	AB	PN	PP	VB	EOS
BOS	1/11	1/10	1/12	1/11	1/25
AB	1/11	1/11	1/11	1/10	1/14
PN	1/11	1/12	1/12	1/10	1/16
PP	1/13	1/11	1/12	1/14	1/18
VB	1/11	1/10	1/10	1/13	1/15

	she	got	up
AB	1/25	1/25	1/14
PN	1/13	1/25	1/25
PP	1/25	1/25	1/13
VB	1/25	1/14	1/19

Set up a fraction containing concrete numbers for the probability that this model assigns to the following tagged sentence. You do not have to simplify.

she	got	up
PN	VB	AB

c) Name two advantages of greedy tagging with the multi-class perceptron over tagging with hidden Markov models based on the Viterbi algorithm.

Sample answers:

a) (i) $43/(0 + 7 + 43) = 43/50$, (ii) $58/(58 + 5 + 0) = 58/63$

b) $\frac{1}{10} \cdot \frac{1}{13} \cdot \frac{1}{10} \cdot \frac{1}{14} \cdot \frac{1}{11} \cdot \frac{1}{14} \cdot \frac{1}{14}$

c) Advantage 1: Lower runtime. Let m and n denote the number of tags and the length of the input sentence, respectively. Then greedy tagging with the multi-class perceptron runs in time $O(mn)$, whereas the Viterbi algorithm needs time $O(m^2 n)$. Advantage 2: Flexibility in feature engineering: The multi-class perceptron can use any kind of hand-crafted features defined over the feature window, whereas hidden Markov models are restricted to transition and emission probabilities.

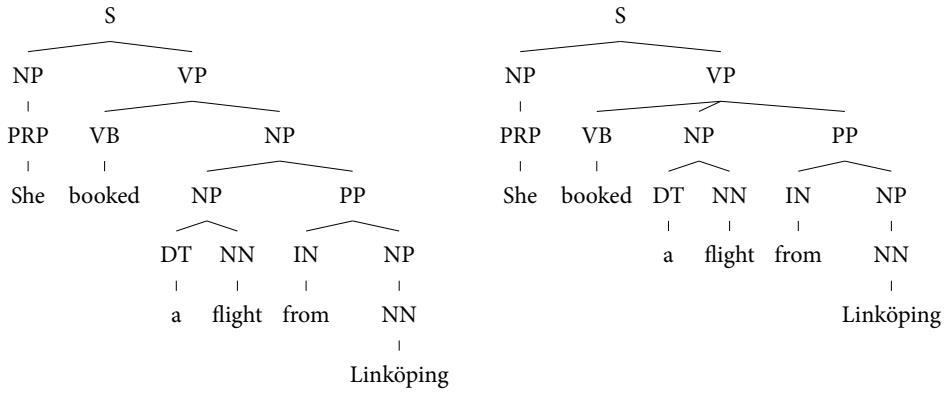
04 **Syntactic analysis** **(3 points)**

a) Here are all NP-rules and all VP-rules together with their probabilities from a certain probabilistic context-free grammar. State the missing numbers a, b, c.

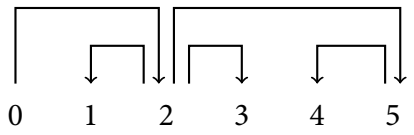
$$\text{NP} \to \text{PRP}\ \tfrac{2}{7} \qquad \text{NP} \to \text{NP PP}\ \tfrac{1}{7} \qquad \text{NP} \to \text{DT NN}\ \tfrac{2}{7} \qquad \text{NP} \to \text{NN}\ a$$

$$\text{VP} \to \text{VB NP}\ \tfrac{1}{b} \qquad \text{VP} \to \text{VB NP PP}\ \tfrac{1}{c}$$

b) Here are two trees that are licensed by the grammar from the previous item. Set up fractions for their probability values. Assume that all rules that are not given above have probability 1.



c) State a sequence of transitions that make an transition-based dependency parser produce the following dependency tree:



Sample answers:

a) $a = \frac{2}{7}, b = 2, c = 2$

b) left tree: $\frac{2}{7} \cdot \frac{1}{2} \cdot \frac{1}{7} \cdot \frac{2}{7} \cdot \frac{2}{7} = \frac{8}{4802}$; right tree: $\frac{2}{7} \cdot \frac{1}{2} \cdot \frac{2}{7} \cdot \frac{2}{7} = \frac{8}{686}$

c) SH SH SH LA SH RA SH SH LA RA RA

05 **Semantic analysis** **(3 points)**

a) Choose the correct semantic relation: synonym, antonym, hyponym, hypernym?

pigeon	is a/an … of	animal
big	is a/an … of	large
parent	is a/an … of	child
begin	is a/an … of	start
screwdriver	is a/an … of	tool

b) Here are three signatures (glosses and examples) from Wiktionary for different senses of the word *course*:

1 A normal or customary sequence. **2** A learning program, as in university. *I need to take a French course.* **3** The direction of movement of a vessel at any given moment. *The ship changed its course 15 degrees towards south.*

Based on these signatures, which of the three senses of the word *course* does the Lesk algorithm predict in the following sentence? Ignore the word *course*, punctuation, and stop words.

In the United States, the normal length of a course is one academic term.

c) We read off word vectors from the following co-occurrence matrix (target words correspond to rows, context words correspond to columns):

	HuSHa'	*Ha'DIbaH*
qa'vIn	5	1
qurgh	5	5
jonta'	1	0
Dargh	1	4

If semantic similarity is measured as the angle between the word vectors, which of the three words below the bar (*qurgh*, *jonta'*, *Dargh*) is most similar to the word above the bar (*qa'vIn*)?

Sample answers:

a) Semantic relations:

pigeon	is a hyponym of	animal
big	is a synonym of	large
parent	is an antonym of	child
begin	is a synonym of	start
screwdriver	is a hyponym of	tool

b) Sense 1 (match with *normal*)

c) The word most similar to *qa'vIn* is *jonta'*. To see this, we draw the four vectors in a two-dimensional coordinate systems with coordinates *HuSHa'* and *Ha'DIbaH*: