Exam 2021-03-19

Student: Sample solutions (version 2)

This exam consists of two parts:

Part A consists of 5 items, each worth 3 points. These items test your understanding of the basic algorithms that are covered in the course. They require only compact answers, such as a short text, calculation, or diagram.

Part B consists of 3 items, each worth 6 points. These items test your understanding of the more advanced algorithms that are covered in the course. They require detailed and coherent answers with correct terminology.

Note that surplus points in one part do not raise your score in another part.

Grade requirements 729G17:

- Grade G: at least 12 points in Part A
- Grade VG: at least 12 points in Part A and at least 14 points in Part B

Grade requirements TDP030:

- Grade 3: at least 12 points in Part A
- Grade 4: at least 12 points in Part A and at least 7 points in Part B
- Grade 5: at least 12 points in Part A and at least 14 points in Part B

Permitted aids: No restrictions on the permitted materials. When using external sources other than the course materials (literature, slides), you must appropriately acknowledge these sources. This also applies to materials obtained from the Internet. You are not allowed to communicate with other people while working on this exam.

Submission instructions: Write each answer on a separate sheet of paper. Photograph or scan each sheet. Combine the different photographs/scans into a single PDF using an online service such as `combinepdf.com`. Submit the final PDF.

Good luck!

Part A

Note: When instructed to 'answer with a fraction', you should provide a fraction containing concrete numbers and standard mathematical operations, but no other symbols. You do not need to simplify the fraction.

like this: $\frac{42+13}{100}$ not like this: $\frac{\#(a)+\#(b)}{N}$

01 Text classification (3 points)

a) Use Maximum Likelihood estimation with add-one smoothing to estimate the class probabilities and word probabilities of a Naive Bayes text classifier from the following document collection. Answer with fractions.

document	class
athens	X
berlin	Y
berlin cairo	Z
cairo athens	Z

b) Based on the probabilities just estimated, compute the class-specific scores that the Naive Bayes classifier uses to predict the class for the following document:

athens berlin cairo

Answer with fractions.

c) Given the following class frequencies, what is the accuracy of the most frequent class baseline on the test data?

Class A	Class B	Class C
1,972	2,958	2,465

(training data)

Class A	Class B	Class C
492	615	738

(test data)

Answer with a fraction.

Sample answers:

a) Class probabilities: $P(\mathsf{X}) = \frac{1}{4}$, $P(\mathsf{Y}) = \frac{1}{4}$, $P(\mathsf{Z}) = \frac{2}{4}$. Word probabilities:

$$P(\text{athens} \mid \mathsf{X}) = \frac{2}{4} \qquad P(\text{athens} \mid \mathsf{Y}) = \frac{1}{4} \qquad P(\text{athens} \mid \mathsf{Z}) = \frac{2}{7}$$
$$P(\text{berlin} \mid \mathsf{X}) = \frac{1}{4} \qquad P(\text{berlin} \mid \mathsf{Y}) = \frac{2}{4} \qquad P(\text{berlin} \mid \mathsf{Z}) = \frac{2}{7}$$
$$P(\text{cairo} \mid \mathsf{X}) = \frac{1}{4} \qquad P(\text{cairo} \mid \mathsf{Y}) = \frac{1}{4} \qquad P(\text{cairo} \mid \mathsf{Z}) = \frac{3}{7}$$

b) Class-specific scores:

$$\begin{aligned}\text{score}(\mathsf{X}) &= P(\mathsf{X}) \cdot P(\text{athens} \mid \mathsf{X}) \cdot P(\text{berlin} \mid \mathsf{X}) \cdot P(\text{cairo} \mid \mathsf{X}) \\ &= \frac{1}{4} \cdot \frac{2}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{2}{256} = \frac{1}{128}\end{aligned}$$

$$\begin{aligned}\text{score}(\mathsf{Y}) &= P(\mathsf{Y}) \cdot P(\text{athens} \mid \mathsf{Y}) \cdot P(\text{berlin} \mid \mathsf{Y}) \cdot P(\text{cairo} \mid \mathsf{Y}) \\ &= \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{2}{4} \cdot \frac{1}{4} = \frac{2}{128} = \frac{1}{128}\end{aligned}$$

$$\begin{aligned}\text{score}(\mathsf{Z}) &= P(\mathsf{Z}) \cdot P(\text{athens} \mid \mathsf{Z}) \cdot P(\text{berlin} \mid \mathsf{Z}) \cdot P(\text{cairo} \mid \mathsf{Z}) \\ &= \frac{2}{4} \cdot \frac{2}{7} \cdot \frac{2}{7} \cdot \frac{3}{7} = \frac{24}{1372} = \frac{6}{343}\end{aligned}$$

c) The most frequent class in the training data is B. Always predicting that class yields a test set accuracy of $\frac{1}{3}$.

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

The WikiText language modelling dataset is a collection of 2 million tokens extracted from verified 'Good' and 'Featured' articles on Wikipedia, comprising a vocabulary of 33,000 unique words. In this dataset we have the following selected counts of unigrams and bigrams:

the	*book*	*the book*	*book the*	*first book*
113,161	611	200	1	8

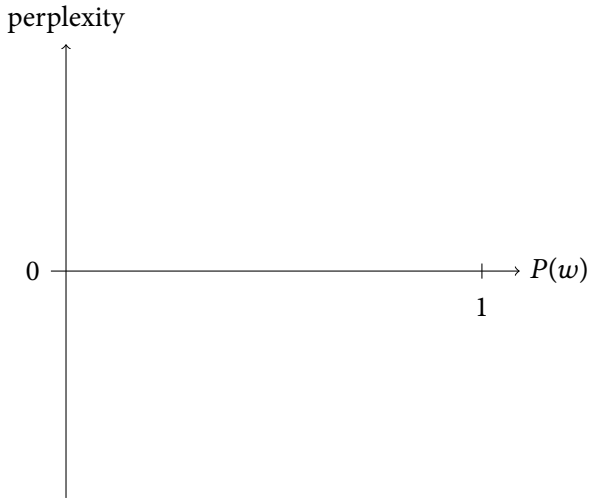
a) Estimate the following probabilities using maximum likelihood estimation without smoothing. Answer with fractions.

i. $P(the)$ ii. $P(book \mid the)$

b) Estimate the following probabilities using maximum likelihood estimation with additive smoothing, $k = 0.1$. Answer with fractions.

i. $P(the)$ ii. $P(book \mid the)$

c) We evaluate a unigram language model on a one-word sentence w. Sketch how the perplexity of the model varies with $P(w)$ by completing the following diagram. What is the minimal value for the perplexity measure?



Sample answers:

a) Maximum likelihood estimation without smoothing:

$$P(the) = \frac{113161}{2 \cdot 10^6}$$
$$P(book \mid the) = \frac{200}{113161}$$

b) Maximum likelihood estimation with additive smoothing, $k = 0.1$:

$$P(the) = \frac{113161 + 0.1}{2 \cdot 10^6 + 0.1 \cdot 33000}$$

$$P(book \mid the) = \frac{200 + 0.1}{113161 + 0.1 \cdot 33000}$$

c) The graph is similar to the one on slide 35 from Lecture 2, except that perplexity grows faster than entropy, and that the smallest possible value is $2^0 = 1$.

03 Part-of-speech tagging (3 points)

a) The evaluation of a part-of-speech tagger produced the confusion matrix shown below. 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`).

	DT	JJ	NN	VB
DT	1030	0	1	2
JJ	2	1325	42	10
NN	0	23	4448	32
VB	1	4	41	3525

Set up fractions for the following values:

i. recall with respect to `NN`

ii. precision with respect to `DT`

b) Training a Hidden Markov model amounts to estimating two types of probabilities. What is the total number of probability values that we need to estimate when training a Hidden Markov model with 16 tags and a vocabulary consisting of 35,623 unique words? Answer with a concrete number. Ignore the beginning-of-sentence and end-of-sentence markers.

c) Here are four statements about the two taggers that we went through in class: the Hidden Markov tagger and the perceptron tagger. Which (zero or more) of these statements are unconditionally true? List the corresponding numbers.

1 There is no guarantee that the perceptron tagger finds the globally optimal assignment of tags to words.

2 The perceptron tagger can achieve a higher tagging accuracy than the Hidden Markov tagger.

3 The Hidden Markov tagger labels each word with that tag which has the highest probability, given the previous word.

4 In contrast to the runtime of the Hidden Markov tagger, the runtime of the perceptron tagger does not grow with the number of tags.

Sample answers:

a) i. $\frac{4448}{0+23+4448+32} = \frac{4448}{4503}$ ii. $\frac{1030}{1030+2+0+1} = \frac{1030}{1033}$

b) 570,224 (256 transition probabilities; 569,968 output probabilities)

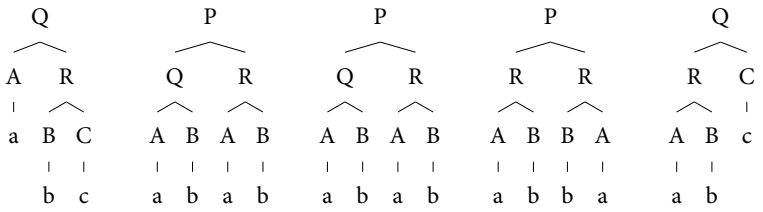
c) 1, 2

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

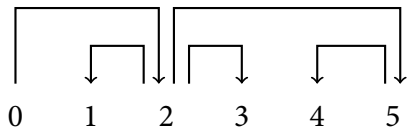
a) You sum up all rule probabilities in a certain probabilistic context-free grammar. Which (zero or more) of the following values are possible as a result?

i. 0.16 ii. 1 iii. 1.6 iv. 1610

b) Below is a small phrase structure treebank. Read off all rules whose left-hand sides are either Q or R and estimate their rule probabilities using maximum likelihood estimation (no smoothing).



c) State two different sequences of transitions that make the transition-based dependency parser produce the following dependency tree:



Sample answers:

a) ii, iv

b) All rules whose left-hand sides are either Q or R:

$$Q \rightarrow A\ R\ \tfrac{1}{4} \quad Q \rightarrow A\ B\ \tfrac{2}{4} \quad Q \rightarrow R\ C\ \tfrac{1}{4}$$
$$R \rightarrow B\ C\ \tfrac{1}{6} \quad R \rightarrow A\ B\ \tfrac{4}{6} \quad R \rightarrow B\ A\ \tfrac{1}{6}$$

c) Possible transition sequences:

- SH SH SH SH RA SH SH LA RA LA RA
- SH SH SH SH RA LA SH SH LA RA RA
- SH SH SH LA SH RA SH SH LA RA RA

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

a) For each of the following pairs of sentences, what is the semantic relation between the emphasized words? Use the correct terminology.

i. A successful musical must have at least three good *songs*. / The crowd cheered after the two teams' national *anthems* had been played.

ii. Ask the band to *play* Waltzing Matilda. / The new *play* premiered today.

iii. The janitor opened the doors to the *school*. / The city will open a new *school* next year.

iv. Steel is a very *strong* material. / The company sells *tough* rucksacks for climbers.

b) Draw a partial WordNet-hierarchy for the following synsets:

1	eggshell	4	fish scale
2	bark	5	rock, stone
3	natural covering, cover	6	natural object

c) Here are five signatures (glosses and examples) from Wiktionary for different senses of the word *green*:

1 Having green as its color. *The former flag of Libya is fully green.* **2** Sickly, unwell. *Sally looks pretty green – is she going to be sick?* **3** Unripe, said of certain fruits that change color when they ripen. **4** Inexperienced. *John's kind of green, so take it easy on him this first week.* **5** Environmentally friendly.

Based on these signatures, which (one or several) of the five senses of the word *green* does the Lesk algorithm predict in the following sentence? Ignore the word *green*, stop words, and punctuation.

He was a young lad, very green and very immature, but friendly.

Sample answers:

a) semantic relations:

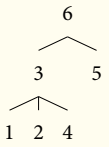
i. hyponymy/hypernymy

ii. homonymy

iii. polysemy

iv. synonymy

b) Partial WordNet-hierarchy for the given synsets:



c) sense 5 (match with word *friendly*)

Part B

06 **Minimum edit distance** **(6 points)**

a) Define the concept of the Levenshtein distance between two words. The definition should be understandable even to readers who have not taken this course.

b) Compute the Levenshtein distance between the two words *student* and *teacher* using the Wagner–Fischer algorithm. Your answer should explicitly state the distance and also show the complete matrix.

c) You have collected a corpus of user-written text, together with a manually spelling-corrected version of the same text. Sketch how this data could be used to learn a corpus-specific version of the Levenshtein distance.

07 **Viterbi algorithm** **(6 points)**

Here is a Hidden Markov model specified in terms of costs (negative log probabilities). The marked cell gives the transition cost from BOS to ADP.

	ADP	ADV	PRON	VERB	EOS
BOS	12	11	10	11	23
ADP	12	13	11	14	18
ADV	11	11	11	10	14
PRON	12	11	12	10	16
VERB	10	11	10	13	15

	she	got	up
ADP	23	23	13
ADV	22	22	14
PRON	13	23	23
VERB	23	14	19

When using the Viterbi algorithm to calculate the least expensive (most probable) tag sequence for the sentence 'she got up', one gets the following matrix. Note that the matrix is missing some values.

		she	got	up
BOS	0			
ADP		35	58	
ADV		33	56	
PRON		23	58	
VERB		34	47	
EOS				

a) Calculate the missing values (the last two columns).

b) Let m and n denote the number of tags in the HMM and the number of words in the input sentence, respectively. The memory required by the Viterbi algorithm is in $O(mn)$, and the runtime required is in $O(m^2n)$. Explain what these statements mean and how to derive them.

c) When one is only interested in the *cost* of the least expensive tag sequence, not in the sequence itself, then the memory required by the Viterbi algorithm is in $O(m)$. Explain this statement. Why does this statement not hold if one wants to reconstruct the actual tag sequence?

08 **Parsing probabilistic context-free grammars** **(6 points)**

Here is a fragment of a probabilistic context-free grammar (PCFG). It is specified here in terms of costs (negative log probabilities) instead of regular probabilities:

S → NP VP	0,97	Det → the	3,98
NP → Det N	5,23	Det → a	3,98
VP → V NP	6,99	N → meal	20,00
V → includes	13,01	N → flight	16,99

We will use this grammar to parse the following sentence:

the flight includes a meal

a) State the problem solved by the probabilistic extension of the CKY algorithm when applied to this grammar. Your statement should be understandable even to readers who have not taken this course.

b) The CKY algorithm assumes the input PCFG to be in *Chomsky normal form*. Explain what this restriction means, and provide an example of a grammar that does *not* satisfy this restriction.

c) Provide the full probabilistic CKY chart for the example sentence. Note that instead of *multiplying* probabilities, you should now *add* their corresponding costs. The other operations of the CKY algorithm remain unchanged.