Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-08-30
Sal (1)	TER3(6)
Tid	14-18
Utb. kod	729G17
Modul	TEN3
Utb. kodnamn/benämning **Modulnamn/benämning**	Språkteknologi Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	8
Jour/Kursansvarig Ange vem som besöker salen	Marco Kuhlmann
Telefon under skrivtiden	013-284644
Besöker salen ca klockan	endast telefonjour
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson, 013-285634, veronica.kindeland.gunnarsson@liu.se
Tillåtna hjälpmedel	inga
Övrigt	
Antal exemplar i påsen	

729G17/729G34/TDP030 Language Technology (2019)

Exam 2019-08-30

Marco Kuhlmann

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 729G17/729G34:

- 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

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) You want to build a text classifier that decides whether a document contains sensitive patient data, such as the patient's name or address. Which of the evaluation measures for text classifiers that you have encountered do you consider to be the most appropriate one for this situation, and why?

b) Here is a document collection:

	document	class
1	Chinese Beijing Chinese	C
2	Chinese Chinese Shanghai	C
3	Chinese Macao	C
4	Tokyo Japan Chinese	J

Use maximum likelihood estimation (without smoothing) to estimate the following probabilities of a Naive Bayes text classifier on this document collection. Answer with fractions.

i. $P(\mathtt{J})$ ii. $P(\text{Chinese} \mid \mathtt{C})$

c) Here is the classification rule for the Naive Bayes classifier, stated using standard probabilities. Re-state it using log probabilities. Explain the notation.

$$\hat{c} = \operatorname*{arg\,max}_{c \in C} \left(P(c) \cdot \prod_{w \in V} P(w \mid c)^{\#(w)} \right)$$

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. In this corpus we have the following counts of unigrams and bigrams:

snow	*white*	*white snow*	*purple*	*purple snow*
38,186	256,091	122	11,218	0

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

i. $P(snow)$ ii. $P(snow \mid white)$

b) Estimate the following probabilities using maximum likelihood estimation with additive smoothing, $k = 0.1$. Assume that the vocabulary consists of 1,254,100 unique words. Answer with fractions.

i. $P(snow)$ ii. $P(snow \mid purple)$

c) We use maximum likelihood estimation with add-k smoothing to train n-gram models on the COCA corpus, with $n \in \{1, \dots, 5\}$ and $k \in \{0, 0.1, 1\}$. The following table shows the entropy of each trained model on the training data. Which row corresponds to which k-value, and why? Answer with a short text.

	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
row 1	7.3376	5.9834	6.7332	6.9556	7.0555
row 2	7.3376	7.3837	8.4573	8.6577	8.7350
row 3	7.3376	3.4269	1.4290	0.5436	0.4171

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 classified a document as class NN whereas the gold-standard class for the document was JJ.

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

Set up fractions for the following values:

i. precision with respect to class VB

ii. recall with respect to class JJ

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

State the probabilities of the following candidate tag sequences for the three-word sentence 'she got up'. Answer with fractions.

PN VB AB   PN VB PP

Part c) on the next page

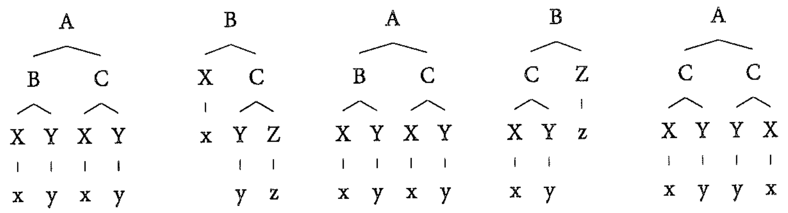
c) The following table shows a comparison between hidden Markov models and multiclass perceptrons. Complete the three missing cells in this comparison.

Hidden Markov model	Multiclass Perceptron
A	scores candidate solutions using weights (real numbers)
B	linear-time greedy search with locally optimal decisions
features: current word, previous word's tag	**C**

04 Syntactic analysis

(3 points)

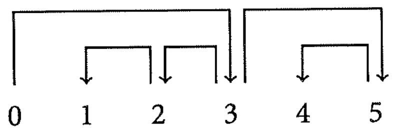
a) Below is a small phrase structure treebank. Read off all rules whose left-hand sides are either B or C and estimate their rule probabilities using maximum likelihood estimation (no smoothing).



b) You sum up all rule probabilities in a certain probabilistic context-free grammar. Which (zero or more) of the following values can you *not* get as a result, and why? Answer with a short text.

i. 0.42 ii. 1 iii. 4.2 iv. 42

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



05 Semantic analysis (3 points)

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

1 car, automobile

2 vehicle

3 self-propelled vehicle

4 dog sled, dog sleigh

5 locomotive

6 sled, sledge, sleigh

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.*

Provide a (grammatical, no-nonsense) sentence containing the word *course* and such that the Simplified Lesk algorithm predicts sense 3 for this word. Ignore the word *course* itself, stop words, and punctuation.

c) We have seen that semantic similarity can be quantified in terms of cosine similarity of word vectors in a high-dimensional vector space constructed from co-occurrence counts. Based on this idea, which (zero or more) of the following words do you expect to end up close to the word *runs* in such a vector space? Motivate your answer with a short text.

i. train   ii. apple   iii. ran   iv. swim

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 *game* and *lake* using the Wagner–Fischer algorithm. Your answer should contain both the distance itself and the complete matrix.

c) Jurafsky and Martin (2018) explain how to augment the Wagner–Fischer algorithm to store backpointers in each cell. Add these backpointers to your matrix. Explain why cells may have several backpointers.

07 Viterbi algorithm (6 points)

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

	PL	PN	PP	VB	EOS
BOS	11	2	3	4	19
PL	17	3	2	5	7
PN	5	4	3	1	8
PP	12	4	6	7	9
VB	3	2	3	3	7

	they	freak	out
PL	17	17	4
PN	3	19	19
PP	19	19	3
VB	19	8	19

When using the Viterbi algorithm to calculate the least expensive (most probable) tag sequence for the sentence 'they freak out' according to this model, one gets the following matrix. Note that the matrix is missing some values.

		they	freak	out	
BOS	0				
PL		28	27	A	
PN		5	28	35	
PP		22	27	B	
VB		23	14	36	
EOS					C

a) Calculate the missing values (marked cells).

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 Named entity tagging (6 points)

Named entity tagging is the task of identifying entities such as persons, organisations, and locations in running text. One way to approach this task is to use the same techniques as in part-of-speech tagging. However, a complicating factor is that named entities can span more than one word. Consider the following sentence:

Alfred Nobel was an inventor from Sweden.

In this example, while the unigram 'Sweden' corresponds to one named entity of type 'location' (LOC), we would also like to identify the bigram 'Alfred Nobel' as a mention of *one* named entity, of type 'person' (PER).

a) To account for the fact that named entities can span more than one word, we can use the so-called IOB tagging. Explain how this scheme works and show how the example sentence given above would be tagged using IOB tagging.

b) Named entity taggers often use *gazetteers*. Explain what a gazetteer is and how it can be integrated into a named entity tagger based on the multi-class perceptron.

c) Jurafsky and Martin (2018) note that evaluation of named entity tagging can be problematic: 'For example, a system that labeled *American* but not *American Airlines* as an organization would cause two errors, a false positive for O and a false negative for I-ORG.' Provide a complete sentence together with gold-standard annotations and system output that illustrates this problem. What accuracy does the system have at the word level? What accuracy does it have at the entity level?