INF4820: Algorithms for AI and NLP
— Sample Exam —

General Instructions

- Please read through the complete exam once before starting to answer questions. About thirty minutes into the exam, the instructor will come around to answer any questions of clarification (including English terminology).
- As discussed in class, the exam is only given in English, but you are free to answer in any of *Bokmål*, English, or *Nynorsk*.
- To give you an idea about the relative weighting of different questions during grading, we've assigned points to each of them (summing to 100).

1 Linear Structures (25 points total)

(a) What is the formula for estimating n-gram probabilities from a training corpus of running text (assuming a straightforward maximum likelihood estimate)? Show the calculations for uni- and bi-grams. *[5 points]*

(b) Assume the following part-of-speech tagged training 'corpus' of just one sentence:

still	,	time	s	move	is	being	received	well	,	once	again	.
RB	,	NNP	POS	NN	VBZ	VBG	VBN	RB	,	RB	RB	.

In a few sentences, discuss the concept of smoothing and explain why it is important. Next, ignoring smoothing and making the standard simplifying assumptions for a naïve bi-gram HMM (including the assumption that the training corpus provides the full inventory of distinct tags and complete vocabulary), calculate the following:

(i) For each tag t, the probability of t following the tag RB, i.e. $P(t|\text{RB})$

(ii) The emission probabilities $P(move|\text{NNP})$, $P(move|\text{NN})$, and $P(well|\text{RB})$.

[10 points]

(c) In a few sentences, summarize the key points of the Viterbi algorithm. What is the interpretation of each cell in the trellis? What is the complexity of the algorithm, i.e. the number of computations performed in relation to (i) the length of the input sequence and (b) the size of the tag set? Very briefly, sketch an alternative, naïve method for computing the most probable tag sequence t_1^n, given an input string w_1^n; state how the Viterbi algorithm improves over this approach. *[10 points]*

2 Classification and Clustering (30 point total)

(a) An important component of vector space models is the metric used for measuring distance or proximity. Give some examples of common metrics and discuss how they differ. (Try to describe the metrics in several ways if you can; e.g., show the formula, illustrate graphically, and offer an intuitive interpretation.) How are the different metrics affected by length normalization? *[10 points]*

(b) The output of bottom-up clustering can be graphically represented using a tree structure known as a dendrogram. Describe what is represented in this structure. *[5 points]*

(c) Explain the concept of *decision boundaries* in relation to classification. *[5 points]*

(d) Rocchio and kNN are two examples of vector space classifiers. Each can be seen to have certain advantages when compared to the other. Compare and discuss the respective merits of the two methods. *[10 points]*

3 Parsing with Context Free Grammars (30 point total)

Consider the language defined by the following grammar (assuming, by convention, 'S' as the start symbol):

S → NP VP	NP → *kim*
VP → VP PP	NP → *oslo*
NP → NP PP	NP → *snow*
VP → V NP	V → *adores*
VP → V	V → *snores*
PP → P NP	P → *in*

(a) For each of the following sentences, identify the number of readings (distinct analyses) that this grammar will assign, and draw the parse trees for each of the readings. *[6 points]*

(i) *kim in oslo adores snow in oslo*

(ii) *kim snores in oslo*

(iii) *kim snores snow in oslo*

(b) Does this grammar contain rules that would be problematic for our naïve, top-down parser? If so, identify the(se) rule(s) and what exactly would constitute a problem for the top-down parser? Discuss the properties of this grammar that either support or prevent the use of a CKY parser. *[7 points]*

(c) Recall very briefly the role of the chart in the CKY, Earley and generalised chart parsers. What types of information are associated with each edge in the chart? For generalised chart parsing, comment on the difference between active and passive edges, and describe the general form of the *fundamental rule*. *[10 points]*

(d) A PCFG can provide us with ranked list of parsed licensed by a given grammar for a given string. In just a couple of sentences, what benefits (if any) could there be in applying a discriminative maximum entropy model fo re-ranking this list? *[7 points]*

4 Common Lisp (15 point total)

(a) The notion of *higher-order functions* plays a central role in Lisp. Explain what we mean by this. *[5 points]*

(b) Define a recursive function named `flat` that takes a possibly nested list as input and returns all of its atomic non-list elements in a flat list. *[6 points]*

For example, for a list like

`(setq foo '(a b c) (((d) (e)) (f (g (h))) i)))`

we expect the following behavior;

`(flat foo)` → `(a b c d e f g h i)`

(c) When evaluating the following sequence of S-expressions, what would be the value returned by the last expression? *[4 points]*

```
(setq foo 42)

(defun foo (x)
  #'(lambda (y)
      (if (>= y x)
          y
        x)))

(let ((bar (foo foo))
      (foo (* foo 2)))
  (funcall bar foo))
```