General instructions

This exam consists of five sections, one for each topic in the course. Each section is worth 8 points, meaning you can obtain a total of 40 points in this exam.

Grade requirements

The following table shows how your total points will map to your grade for the exam, depending on your course code:

	0–23	24–26	27–29	30–32	33–35	36–40
729G17	U	G	G	G	VG	VG
729G86	F	E	D	C	B	A
TDP030	U	3	3	4	4	5

The threshold for a passing grade is always 60%, i.e. at least 24 points.

Instructions for mathematical expressions

When a question requires you to answer with a numerical expression, all expressions that evaluate to the correct answer will be considered equally correct. In other words: **You do not need to simplify fractions or evaluate them yourself.**

For example, if the correct answer to a question is $\frac{1}{2}$, then all the following answers will be considered equally correct:

$0.5, \frac{1}{2}, \frac{2}{4}, \frac{0.5}{1}, \frac{5 \times 10}{10^2}$

To write a fraction in Inspera, use the division button (÷) or type a slash (/) on your keyboard. For example, you can write the fraction $\frac{1}{2}$ by typing “1/2” on your keyboard or by pressing the buttons $1 \div 2$ on the Inspera interface.

Tips

- It’s a good idea to **go through the entire exam first** and familiarize yourself with the tasks. This way, if you don’t understand something, you can ask questions early.
- **Solve the easier questions first!** You can always skip questions that you find hard and come back to them later.
- Some questions are **intentionally harder than others**, particularly those that require free-text answers. Remember that you will pass the exam with 60% of available points, so the more difficult questions are only required if you are aiming for a higher grade.

Good luck!

Sample answers are given in either an annotation box (like this one) or marked in green.

1.1 Text classification: MLE

Here is a collection of documents and their gold-standard class labels:

document	class
laugh laugh movie	comedy
hilarious laugh	comedy
movie scary tension	horror

Use **maximum likelihood estimation (MLE) with add-one smoothing** to estimate the following probabilities of a Naive Bayes classifier from the given document collection. Assume that the vocabulary consists of the set of all words occurring in the documents.

a) The class probability of "horror"

$P(\text{horror}) =$ (1/3)

b) The word probability of "movie" given the "comedy" class

$P(movie|\text{comedy}) =$ (1/5) or: (1+1)/(5+1*5)

c) The word probability of "laugh" given the "horror" class

$P(laugh|\text{horror}) =$ (1/8) or: (0+1)/(3+1*5)

Maximum marks: 3

1.2 Text classification: Naive Bayes

A Naive Bayes classifier has been trained on classifying whether an e-mail is spam or not, resulting in the following class probabilities and (some of the) word probabilities:

	class prob.	word prob. for buy	word prob. for money	word prob. for now
spam	$\frac{1}{3}$	$\frac{17}{80}$	$\frac{17}{80}$	$\frac{10}{80}$
not spam	$\frac{2}{3}$	$\frac{1}{120}$	$\frac{2}{120}$	$\frac{13}{120}$

Based on the probability values above, **compute the class-specific scores** that the Naive Bayes classifier uses to predict the class for the following document:

- *money money now*

	estimated probability
spam	((1/3)*(11/80)*(11/80)*(8/80))
not spam	((2/3)*(3/120)*(3/120)*(18/120))

Maximum marks: 2

1.3 Text classification: Most frequent class

Here are some counts of class labels from a dataset:

	A	B	C	D
training data	35	410	355	440
test data	10	74	69	21

Given the counts above, what is the **accuracy** of the **most frequent class baseline** on the test data?

21 / (10+74+69+21)

Maximum marks: 1

1.4 Text classification: Baselines

Your responses in text fields are saved automatically

What is a “baseline”, and why is it important for evaluating text classifiers? Explain in a few sentences.

A baseline is a simple classifier that we use to compare against classifiers we want to test.
It is important to compare against a baseline because evaluation metrics are not absolute measures of performance. (or: because we can only interpret evaluation metrics by comparing them against something else)

Maximum marks: 2

2.1 Language modelling: MLE

Consider a dataset of English fiction, containing a total of 99000 tokens with a vocabulary of 4000 unique words. Assume that the following counts of unigrams and bigrams were extracted from this dataset:

space	*ship*	*space ship*	*ship space*
1023	858	4	0

a) Estimate the following probabilities using maximum likelihood estimation **without smoothing**.

$P(space)$	$P(ship \mid space)$
1023 / 99000	4 / 1023

b) Estimate the following probabilities using maximum likelihood estimation **with add-k smoothing**, using $k = \frac{1}{2}$.

$P(space)$	$P(space \mid ship)$
(1023 + 0.5) / (99000 + 0.5 * 4000)	(0 + 0.5) / (858 + 0.5 * 4000)

Maximum marks: 4

2.2 Language modelling: Entropy and perplexity

A language model computes probability values for sequences of words. However, it is more common to use **entropy and perplexity** when evaluating a language model. Which of the following statements about these metrics are correct?

Mark all correct answers!

- [ ] The perplexity will always be a value ≥ 1. ✔
- [ ] The perplexity of a language model is its average surprisal per word.
- [ ] A high probability value corresponds to a low entropy value. ✔
- [ ] The entropy of a language model is its average surprisal per word. ✔

Maximum marks: 2

2.3 Language modelling: BOS token

Your responses in text fields are saved automatically

In a bigram language model, we introduce special beginning-of-sentence (BOS) and end-of-sentence (EOS) tokens. For example, consider the sentence:

- *my cat loves belly rubs*

In a bigram language model, we might represent this sentence with the following tokens:

- *BOS my cat loves belly rubs EOS*

Why do we introduce the beginning-of-sentence token (BOS)? What problem does that solve? Explain in a few sentences.

A bigram model calculates conditional probabilities of each word given the previous word. We introduce the BOS token so that we can compute a probability for the first word of a sentence, e.g. P(my|BOS).

(Without BOS, we could not assign any probability to the first word of a sentence.)

(You don't need to discuss the EOS symbol in your answer.)

Maximum marks: 2

3.1 Sequence labelling: Precision and recall

Below is a confusion matrix from evaluating a part-of-speech tagger. Rows correspond to gold-standard tags, while columns correspond to predicted tags. *(For example, the highlighted cell contains the number of times the classifier predicted VERB where the gold-standard class was NOUN.)*

	ADJ	DET	NOUN	VERB
ADJ	430	4	82	6
DET	2	854	0	0
NOUN	10	2	2020	106
VERB	14	0	131	1428

Based on the confusion matrix above, compute the following evaluation metrics!

a) precision with respect to ADJ

430 / (430+2+10+14)

b) recall with respect to NOUN

2020 / (10+2+2020+106)

Maximum marks: 2

3.2 Sequence labelling: Span-level precision/recall

Below is a sentence with named entity spans ***highlighted***, both according to a gold-standard annotation and the prediction of a named entity recognition model.

gold-standard *The **University of Michigan** was established in **Ann Arbor** in **1837**.*
predicted ***The University of Michigan*** *was established in **Ann Arbor** in 1837.*

Based on these underlined spans, compute the following **span-level** evaluation metrics!

span-level precision	span-level recall
(1/2)	(1/3)

Maximum marks: 2

3.3 Sequence labelling: Named entity annotation

Below is a sentence with named entity spans ***highlighted*** and annotated with their type. For example, the span “Russian” is annotated as a named entity of type “NAT” (nationality).

Anna Filosofova	*was a*	***Russian***	*feminist.*	
PER		**NAT**		

Convert the named entity annotations shown above to **BIO notation** by choosing the correct tag for each token!

token	tag
Anna	Select alternative (B-PER, I-PER, B-NAT, I-NAT, O)
Filosofova	Select alternative (B-PER, I-PER, B-NAT, I-NAT, O)
was	Select alternative (B-PER, I-PER, B-NAT, I-NAT, O)
a	Select alternative (B-PER, I-PER, B-NAT, I-NAT, O)
Russian	Select alternative (B-PER, I-PER, B-NAT, I-NAT, O)
feminist	Select alternative (B-PER, I-PER, B-NAT, I-NAT, O)

Maximum marks: 2

3.4 Sequence labelling: Perceptron tagger

Which of the following features can be used in a part-of-speech tagger using a **multi-class perceptron**? **Mark all that apply!**

- [x] the word form of the next token ✔
- [x] the word form of the current token ✔
- [x] the last three letters of the current token ✔
- [ ] the part-of-speech tag of the next token
- [x] the first three letters of the next token ✔
- [x] the part-of-speech tag of the previous token ✔

Maximum marks: 2

4.1 Word embeddings: Semantic relations

For each of the following examples, what **semantic relation** best describes the relation between the ***highlighted*** words?

example	semantic relation
*The **loud** music made it hard to hear her **quiet** voice.*	Select alternative (synonymy, hypernymy, antonymy, homonymy)
*He hit the ball with a **bat**. — The **bat** flew out of the cave.*	Select alternative (antonymy, hypernymy, synonymy, homonymy)
*They **talked** and **discussed** for hours.*	Select alternative (hypernymy, synonymy, antonymy, homonymy)
*A **rose** is a beautiful **flower**.*	Select alternative (hypernymy, synonymy, homonymy, antonymy)

Maximum marks: 2

4.2 Word embeddings: Distributional hypothesis

Your responses in text fields are saved automatically

State the "distributional hypothesis" in one or two sentences!

We can learn something about the meaning of a word by looking at its context (= surrounding words).

Maximum marks: 1

4.3 Word embeddings: Cosine similarity

Here are some words in a fictional language, together with their vectors in a two-dimensional embedding space:

word	vector
kireth	$(3, 1)$
lunariq	$(2, 2)$
threvali	$(-1, -2)$
zelvanu	$(-2, 2)$

Sort the words in decreasing degree of similarity (1. = most similar, 4. = least similar) to the word *lunariq*! Assume that similarity is measured using cosine similarity between the word vectors.

Help

1. lunari ✔
2. kireth ✔
3. zelva ✔
4. threv ✔

zelvanu

threvali

lunariq

kireth

Maximum marks: 2

4.4 Word embeddings: Skip-gram with negative sampling

One method to train word embeddings is **skip-gram with negative sampling (SGNS)**, which was implemented in the *word2vec* tool. SGNS trains a simple neural network classifier, and uses the representations from the trained neural network model as the word embeddings.

Briefly explain the main ideas behind skip-gram with negative sampling (SGNS)! Your explanation should include

- what classification task the model is trained on,
- what “negative sampling” means and how it works, and
- why this method would result in “good” word embeddings (i.e., the motivation behind this approach).

(You do *not* need to give any mathematical formulas, and you do *not* need to know or write anything about the learning algorithm for a neural network. Describing the *conceptual* ideas of SGNS is sufficient.)

Format

SGNS is trained on the task of classifying whether a given pair of words appeared in the same context or not (+ or –).

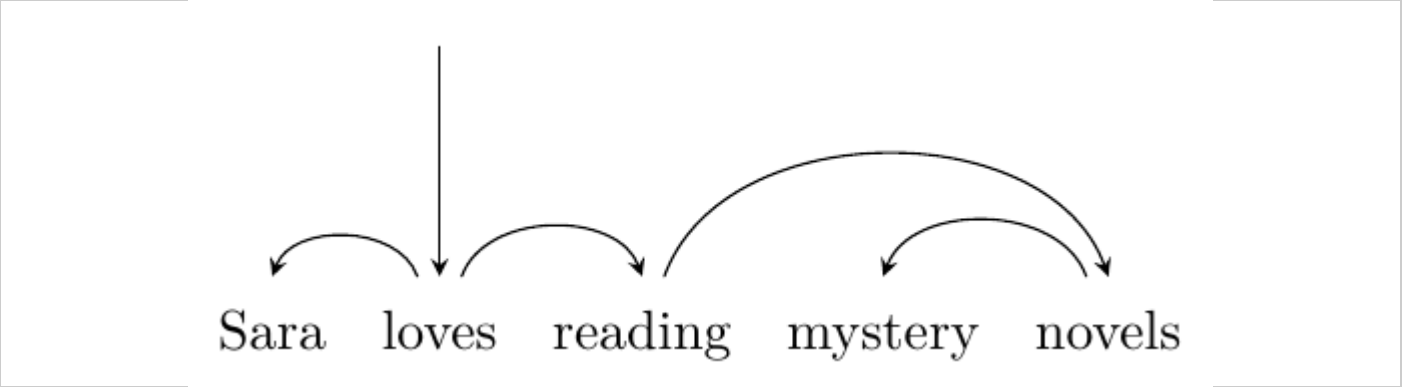
Negative sampling finds training examples for the negative class (–) by randomly sampling words from the entire vocabulary.

Predicting whether or not two words appeared in the same context should result in meaningful word embeddings because the distributional hypothesis says that this tells us something about the meaning of these words.

Words: 0/500

Maximum marks: 3

5.1 Syntactic analysis: Transition-based parsing



Complete the **transition sequence** below (using the arc-standard model) so that it produces the dependency tree shown here!

(Note: There is only one valid transition sequence that produces the tree and starts with "SH SH LA". You will need exactly seven more transitions, so make sure to fill in every gap.)

Help

RA LA SH

SH SH LA S ✔ S ✔ S ✔ L ✔ R ✔ R ✔ R ✔

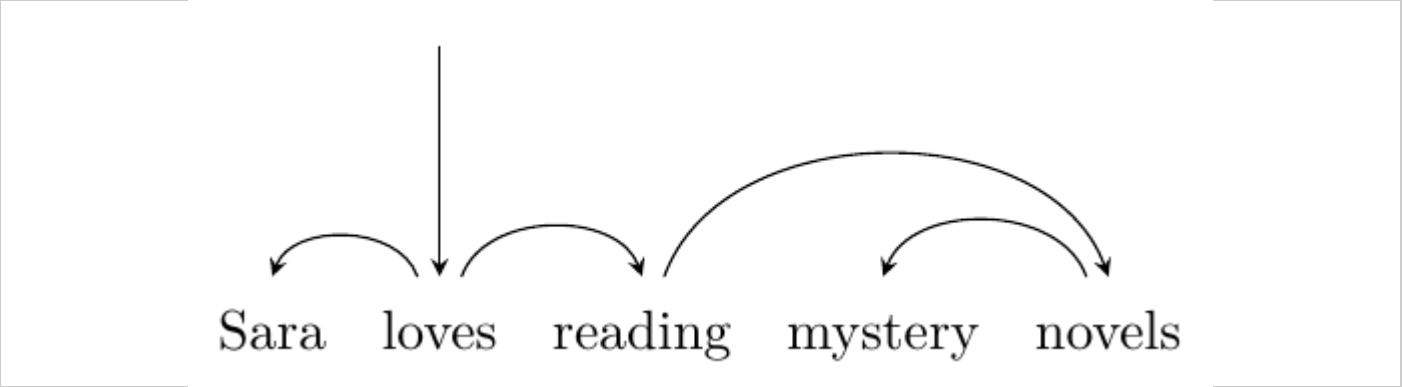
SH SH SH LA RA RA RA

Reminder:

- SH = Shift
- LA = Left arc
- RA = Right arc

Maximum marks: 2

5.2 Syntactic analysis: Heads and dependents

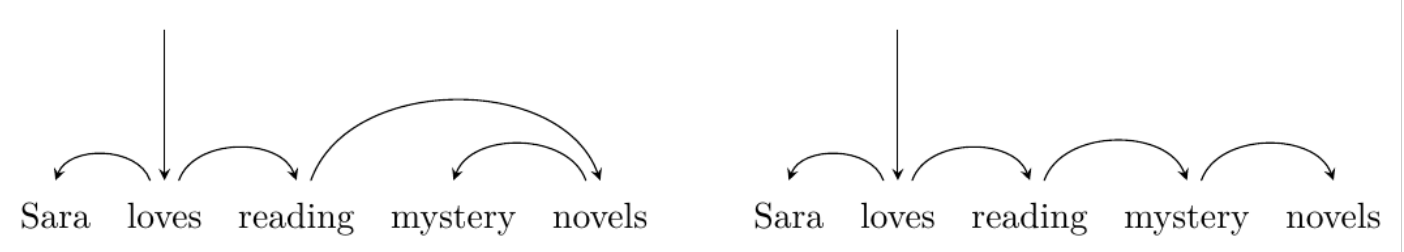


Convert the dependency tree shown here into a **token-level annotation format** (like the one used by CoNLL-U) by filling out the table below. Assume that the tokens are indexed from 1 to 5. Each row should correspond to one arc in the dependency tree, with the "**dependent**" column giving the index of the dependent token, and the "**head**" column giving the index of the head token. Use the index 0 for the special "root" node.

token	dependent	head
Sara	(1)	(2)
loves	(2)	(0)
reading	(3)	(2)
mystery	(4)	(5)
novels	(5)	(3)

Maximum marks: 2

5.3 Syntactic analysis: Attachment scores



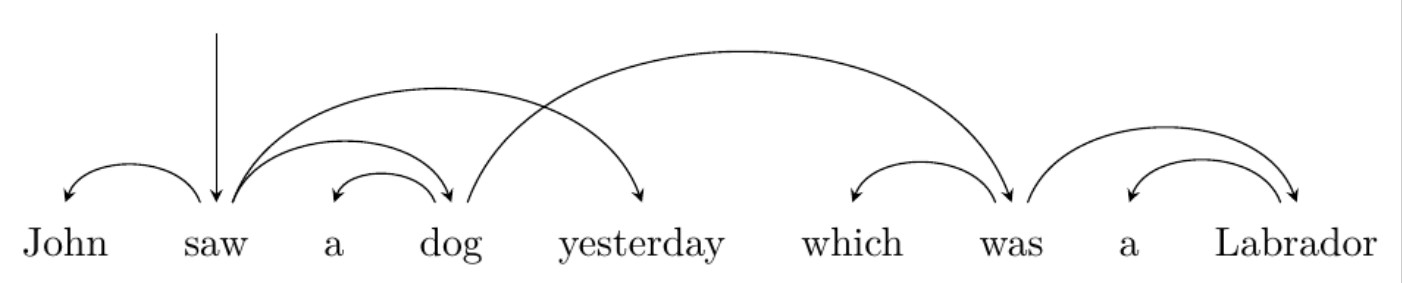
Assume that the dependency tree on the left is the gold-standard tree, while the tree on the right was predicted by a dependency parser.

What is the **unlabelled attachment score (UAS)** of this parser on this sentence?

(3/5)

Maximum marks: 1

5.4 Syntactic analysis: Valid dependency trees



Is the graph shown here a **valid dependency tree?** State your answer and the properties that a valid dependency tree must have, and explain which of these properties the graph does or does not fulfill.

Format

1. Each node should have exactly one incoming edge/arc/arrow.
2. Each node should be reachable from the root node.

Both properties are fulfilled, so this is a valid dependency tree.

(The crossing edges make it non-projective, but that doesn't matter here.)

Words: 0/500

Maximum marks: 3