- **You should answer all questions. The weights of the various exercises are indicated.**
- **You should read through the whole set to see whether anything is unclear so that you can ask your questions to the teachers when they arrive.**
- **If you think some assumptions are missing, make your own and explain them!**

1 Experiments (10%)

What does it mean to perform n-fold cross-validation, e.g. 5-fold cross-vaildation experiments? What are the advantages of using cross-validation?

In a machine learning experiment you split your data into two disjoint parts: training data and test data. You train the machine learner on the training data and test it on the test data.

In n-fold cross-validation you split your data into n equal parts and perform n different experiments. Each of the n different parts serve as test data in one experiment, and in this experiment the $n-1$ other parts are merged and used as training data.

The main advantage is that you test on a larger test material, and therefore get more reliable results. For example, with a data set of 1000 items, instead of, say, training on 800 items and test on 200 items, you may test on 1000 items. For some performance measures, e.g. accuracy, you may simply use the mean of the measures for the n many experiments as a measure.

2 Evaluation (20%)

Kim is not satisfied with the automatic language recognizer installed on her PC. It often mistakes Norwegian Bokmål as other languages. Kim therefore implements a classifier and evaluates it on a test set of 900 sentences. This is the result.

	Correct class		
	Bokmål	Nynorsk	Danish
Assigned Bokmål	240	0	10
Assigned Nynorsk	50	300	20
Assigned Danish	10	0	270

(a) What is the accuracy of this classifier?

Accuracy $= \frac{240+300+270}{900} = 0.9$

(b) Kim is mainly interested in whether the classifier recognizes Bokmål correctly or not. What is the accuracy, precision, recall and F-score for the Bokmål class on the test set?

$$\text{Accuracy} = \frac{240 + 300 + 270 + 20}{900} = \frac{830}{900} = \frac{83}{90} = 0.9222...$$

$$\text{Precision} = \frac{240}{240 + 10} = 0.96$$

$$\text{Recall} = \frac{240}{240 + 50 + 10} = 4/5 = 0.8$$

$$\text{F-score} = \frac{2PR}{P + R} = \frac{2 * \frac{24}{25} * \frac{4}{5}}{\frac{24}{25} + \frac{4}{5}} = \frac{\frac{192}{25*5}}{\frac{24*5+4*25}{25*5}} = \frac{192}{220} = \frac{48}{55} \approx 0.873$$

(c) Returning to the three-class classifier and the accuracy found in question (a). Assume that the test set is a random sample of sentences from a large population of sentences in the three languages. Estimate a confidence interval for the accuracy at the 95% confidence level.

Formula for the interval is

$$[\hat{p} - Z^* \frac{s}{\sqrt{n}}, \hat{p} - Z^* \frac{s}{\sqrt{n}}]$$

Here

$$n = 900$$

$$\hat{p} = 0.9$$

$$s = \sqrt{\hat{p}(1 - \hat{p})} = \sqrt{0.9 \times (1 - 0.9)} = 0.3$$

$$Z^* = 1.96$$

$$[\hat{p} - Z^* \frac{s}{\sqrt{n}}, \hat{p} - Z^* \frac{s}{\sqrt{n}}] = [0.9 - 1.96 \frac{0.3}{\sqrt{900}}, 0.9 + 1.96 \frac{0.3}{\sqrt{900}}]$$

$$= [0.9 - 1.96 \times 0.01, 0.9 + 1.96 \times 0.01]$$

$$\approx [0.088, 0.92]$$

3 Information extraction (20%)

(a) What is meant by NP-chunking? Propose an NP-chunk structure for the following sentence. Use parenthetical notation. You do not have to include the POS-tags.

```
[('American', 'NNP'),
('Petrofina', 'NNP'),
('Inc.', 'NNP'),
(',', ','),
('an', 'DT'),
('integrated', 'VBN'),
('oil', 'NN'),
('company', 'NN'),
('based', 'VBD'),
('in', 'IN'),
('Dallas', 'NNP'),
(',', ','),
('yesterday', 'NN'),
('said', 'VBD'),
('net', 'JJ'),
('income', 'NN'),
('dropped', 'VBD'),
('to', 'TO'),
('$', '$'),
('15.1', 'CD'),
('million', 'CD'),
(',', ','),
('from', 'IN'),
('$', '$'),
('35.2', 'CD'),
('million', 'CD'),
 ('.', '.')]
```

(American Petrofina Inc.) , (an integrated oil company) based in (Dallas) , (yesterday) said (net income) dropped to ($ 15.1 million) , from ($ 35.2 million) .

NP-chunking is the process of finding non-overlapping segments of the sentence corresponding to NPs. It returns a flat non-hierarchical structure. When a phrase-structure will identify one NP, a, as part of another NP, b, the chunk structure may identify a as one NP and a a smaller part of b, excluding a, as another NP. For example, a phrase-structure will identify *Dallas* as one NP and *an integrated oil company* as another NP.

(b) A popular format for representing chunk structure is the so-called BIO (or IBO) tags. Display the chunk structure from (a) using BIO-tags and a CoNLL-type format. Include the POS-tags.

See (d)

(c) What is meant by named entity recognition (NER)? A named entity is anything that can be referred to by a proper name or definite description, like a person, an organization, a location, ... It is also usual to include dates, times and other referring temporal expressions.

Depending on the application at hand, on introduces a set of entity types, typically PER, ORG, LOC, TIME, for all applications, and then e.g., DIAGNOSIS, DRUG, ... if the application is from the medical domain.

Named entity recognition consists in identifying the segments of the sentence that constitute proper names and classify these segments with one of the entity types.

(d) Propose a set of named entity types suitable for the business world with revenues, mergers and acquisitions. Annotate the example sentence with named-entities accordingly by extending the representation from (b).

We include the generic types: PERS, LOC, TIME. We need a type for companies, like *American Petrofina Inc.*. We could use ORG, or we could make it more fine-grained, say separating between commercial companies, and e.g., public/governemental offices, and maybe some more organisations. We also nee to be able to speak of amounts of MONEY.

WORD	POS	NP-Chunk	NE-Label
'American'	'NNP'	B	B-ORG
'Petrofina'	'NNP'	I	I-ORG
'Inc.'	'NNP'	I	I-ORG
','	','	O	O
'an'	'DT'	B	O
'integrated'	'VBN'	I	O
'oil'	'NN'	I	O
'company'	'NN'	I	O
'based'	'VBD'	O	O
'in'	'IN'	O	O
'Dallas'	'NNP'	B	B-LOC
','	','	O	O
'yesterday'	'NN'	B	B-TIME
'said'	'VBD'	O	O
'net'	'JJ'	B	O
'income'	'NN'	I	O
'dropped'	'VBD'	O	O
'to'	'TO'	O	O
'$'	'$'	B	B-MONEY
'15.1'	'CD'	I	I-MONEY
'million'	'CD'	I	I-MONEY
','	','	O	O
'from'	'IN'	O	O
'$'	'$'	B	B-MONEY
'35.2'	'CD'	I	I-MONEY
'million'	'CD'	I	I-MONEY
'.'	'.'	O	O

(e) How is NER performed by supervised machine learning? What kind of features could be useful?

NER is typically considered a word-by-word sequence labelling task where the goal is to choose the right NE-BIO-tag for each word. One may use different sequence classifiers, the most common used to be CRFs.

Typical features can be found from the other columns in the table: word-form, POS-tag and Chunk-tag for the word itself and for neighboring words.

In addition to wordform, it may be useful to consider suffixes and prefixes of various length (say up to 4) for the word.

It can also be useful to consider whether first letter is capital, all letters are capitals etc. for the word and neighboring words.

If one has run other analyses on the sentences, e.g., dependency parsing, features from that may be applied.

Another world of features open up if one includes external knowledge sources, e.g., gazetteers—list of place names—or—for the example sentnece—lists of companies.

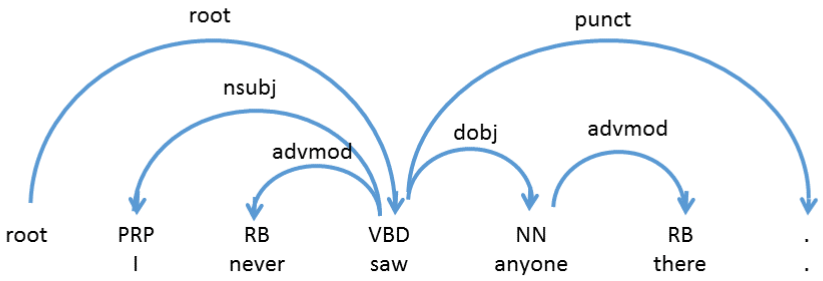
4 Data-driven dependency parsing (20%)

4.1 Dependency trees

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL
1	I	I	PRON	PRP	Case=Nom|Number=Sing	3	nsubj
2	never	never	ADV	RB	_	3	advmod
3	saw	see	VERB	VBD	Tense=Past|VerbForm=Fin	0	root
4	anyone	anyone	PRON	NN	Number=Sing	3	obj
5	there	there	ADV	RB	PronType=Dem	4	advmod
6	.	.	PUNCT	.	_	3	punct

(a) Draw the dependency graph for the English sentence '*I never saw anyone there.*', provided above in the *CoNLL* format.

Should be something similar to the graph below:



(b) Is this sentence projective? Why or why not?

It is projective because the arcs between the nodes do not cross each other.

(c) Which of the dependents of '*saw*' are arguments, and which are adjuncts? Why?

'*I*' and '*anyone*' are arguments, because they are obligatory for the verb '*to see*'. '*Never*' is adjunct, because it expressed an optional property of the 'seeing' action.

4.2 Automatic dependency parsing

(a) Recall any two features commonly used in data-driven dependency parsing and use the dependency tree in 4.1 to give an example for each of the features.

Students can come up with many different features. For example: parts of speech and lemmas of the word pair, like '*see_ VERB*' and '*anyone_ PRON*' in the graph above.

(b) Briefly describe the problems with categorical discrete features for dependency parsing that artificial neural networks somewhat helped to alleviate (and how).

Briefly: discrete features are problematic in that there are to many of them (considering their combinations). Thus, it is difficult to learn proper weighting for these millions of features. Another problem is that categorical features are inherently not related to each other, which means that, for example, the words '*cup*' and '*mug*' for the model are equally distant as '*cup*' and '*ocean*', which is obviously wrong.

Neural networks allow to replace discrete categorical features with their *embeddings* in some low-dimensional space. These embeddings can even be learned by the model itself during training. Thus, statistics is shared between similar words and PoS tags and the model works much faster, because the number of features is dramatically lower.

5 Word sense disambiguation (10%)

(a) Outline the difference between *word sense disambiguation* and *word sense induction*.

Word sense disambiguation (WSD) is the task of determining what sense of a particular word is being used in a particular context (from a predefined sense inventory). Word sense induction (WSI) is the task of discovering sense inventory for a word from text.

(b) What machine learning paradigms are usually associated with the former and the latter?

WSD is usually associated with supervised machine learning, particularly *classification*. WSI is inherently unsupervised task, practically equal to *clustering*.

6 Semantic role labeling (20%)

6.1 Semantic datasets

1	Apparently	apparently	RB	4	adv	_	AM-DIS
2	the	the	DT	3	nmod	_	_
3	commission	commission	NN	4	sbj	_	A0
4	did	do	VBD	0	root	_	_
5	not	not	RB	4	adv	_	AM-NEG
6	really	really	RB	4	adv	_	AM-DIS
7	believe	believe	VB	4	vc	believe.01	_
8	in	in	IN	7	adv	_	A1
9	this	this	DT	10	nmod	_	_
10	ideal	ideal	NN	8	pmod	_	_

(a) The sentence above has been annotated with semantic roles in the *CoNLL08* format. Identify the predicate of this sentence and its semantic arguments along with their roles.

The predicate here is '*believe*'. Its arguments are '*apparently*' (discourse modifier), '*commission*' (A0 or agent), '*not*' (negation modifier), '*really*' (discourse modifier) and '*in*' (A1, object or patient).

(b) Provide a short description for each of the roles of the *core* arguments in this sentence.

The core arguments here are '*commission*' and '*in*'. The first one is the syntactic subject of the sentence. In terms of Dowty's proto-roles it is A, or PROTO-AGENT, the entity performing action, or causing event. '*In*' here serves as a proxy to the clause '*in this idea*l', which has the semantic role of A1 or PROTO-PATIENT, the entity receiving the consequences of the action, something causally affected by the event.

(c) Invent two English sentences describing one event with different dependency trees but identical semantic role structures. Briefly explain the discrepancies between syntax and semantics in them.

For example: '*Jack tricked Jill*' and '*Jill was tricked by Jack*'. Syntactically, they are different: '*Jack*' is the subject in the first sentence, but it serves as oblique in the second sentence. '*Jill*' is an object in the first sentence, but moves to be the (passive) subject in the second one, etc. However, the semantic roles assigned to '*Jack*' and '*Jill*' are exactly the same: '*Jack*' is A0 and Jill is A1. The reason for that is that the event is still the same, and the one causing the action is '*Jack*', while '*Jill*' is receiving its consequences.

6.2 Automatic SRL

(a) Outline the general workflow of a machine learning based SRL system taking raw text as an input.

It is usually something like: 1) syntactic parsing; 2) pruning constituents (removing words which can't be predicates or arguments); 3) identifying arguments and predicates with supervised machine learning; 4) classifying arguments into their semantic roles using supervised ML again (usually generating several hypotheses); 5) re-ranking hypotheses based on linguistic constraints (structural inference).

(b) Choose any *non-core* argument from the sentence in 6.1. Give examples of features which can be helpful in correctly identifying its role.

One example can be like this. Let's take the argument '*commission*'. To identify its role (A0) probably two features will be very helpful: its part of speech (noun) and its dependency relation to the predicate (subject). A noun governed by the predicate with the 'subject' arc has very high chances to become A0.