- **Du kan svare på norsk, dansk, svensk eller engelsk.**
- **Du skal svare på alle spørsmålene. Vekten på de ulike spørsmålene er oppgitt.**
- **Du bør lese gjennom hele settet slik at du kan stille spørsmål til faglærerne når de kommer i tilfelle noe er uklart.**
- **Hvis du føler noen forutsetninger mangler, lag dine egne og redegjør for dem!**

Oppgave 1 (samlet 15%)

Spørsmål 1.1 (5%)

Forklar kort hva som menes med *ordforekomst, ordform, leksem* og *lemma* med utgangspunkt i

1. *mouse, mice, rat, rats*

og teksten

2. *Rats aren't smarter than mice, and that's important. "Anything we could train a rat to do we could train a mouse to do as well", says Tony Zador. This finding is important because using mice in experiments instead of rats could open up all kinds of new research options.*

- Med ***ordforekomst*** sikter vi til ord i tekst (eller tale). I den tredje setningen er *of* på plass 11 en ordforekomst og *of* på plass 18 en annen forekomst, og det er 21 ordforekomster i alt i setningen.
- ***Ordform*** er et leksikalsk begrep. De to forekomstene av *of* er forekomster av samme ordform, og det er 20 forskjellige ordformer i setning 2.
- Formen *rats* er flertallsform og *rat* er entallsform av samme **leksem**. Disse to formene (sammen med evt andre former) utgjør et leksem. Tilsvarende tilhører *mouse* (entallsform) og *mice* (flertallsform) samme leksem.
- Med ***lemma*** mener vi siteringsformen for leksemet. For substantiv i engelsk er det entallsformen, *rat* og *mouse*. Lemmaet for *mice* er *mouse*.

Spørsmål 1.2 (5%)

Hva er "stemming" og hva er lemmatisering? Hva er forskjellene mellom de to?

For en del oppgaver som f.eks. søk ønsker vi ikke å skjelne mellom ulike former av et leksem. Vi ønsker at et søk på *mouse* også skal gi et tilslag på den første setningen fordi den inneholder *mice*. Lemmatisering vil si å skifte ut ord med tilsvarende lemma, altså *rat* og *rats* med *rat*, og *mouse* og *mice* med *mouse*. Lemmatisering krever at systemet har informasjon om at f.eks. *mice* er flertallsform av *mouse*. Stemming er en enklere regelbasert prosess som forsøker å fjerne endelser fra ord og dermed oppnå at ord fra samme leksem får samme stamme. En stemmer vil kunne redusere både *rat* og *rats* til *rat*, men den vil ha problemer med å redusere *mouse* og *mice* til det samme. Det eneste de to formene deler er første bokstav.

Spørsmål 1.3 (5%)

Hva mener vi med tokenisering ("tokenization") i språkteknologi? Med utgangspunkt i teksten (2) over, diskuter kort valg som kan oppstå når en skal lage en tokeniserer.

En elektronisk lagret tekst i rå form er en sekvens av tegn (eng. "characters"). Tokenisering vil si å dele denne opp i enheter som tilsvarer ord. I tredje setningen vil f.eks. *This* bli første token, *finding* andre token, *is* tredje token osv.

Den enkleste er å dele opp tegnsekvensen ved blanke (eng.: "white space"). Første spørsmål som oppstår er hva vi skal gjøre med skilletegn ved slutten eller begynnelsen av ord, f.eks. *mice,* og *important.* i setning 1. Her er det flere alternativ, i hvert fall disse:

- Skille mellom ord og skilletegn og la dem være hver sin token, altså en token *important* og en annen for punktum.
- Fjerne skilletegnet og bare ha en token, *important*
- La skilletegnet forbli på ordet, og bare ha en token, *important.*

Et annet spørsmål er hva vi gjør med sammentrekninger som *aren't* og *That's*. Også her er det flere muligheter, som

- beholde *aren't* som en token
- splitte før ', slik at det blir to tokens *aren* og *'t*
- splitte i *are* og *n't*
- splitte og normalisere til *are* og *not*.

Et tredje spørsmål er om vi skal gjøre om store bokstaver til små i begynnelsen av ord. F.eks. er *Rats* i begynnelsen av den første setningen en forekomst av samme ordform som *rats* i siste setningen. Skal vi i så fall bare gjøre om for første ord i setningen eller for alle ord?

(Enda et spørsmål, som ikke viser seg i teksten: Sammensatte ord ("compounds") i engelsk skrives noen ganger som to ord *data base* og andre ganger som ett ord, *database*. Et spørsmål er om dette bør være én token også når det er en blank mellom ordene, altså *data base* som en token eller som to.)

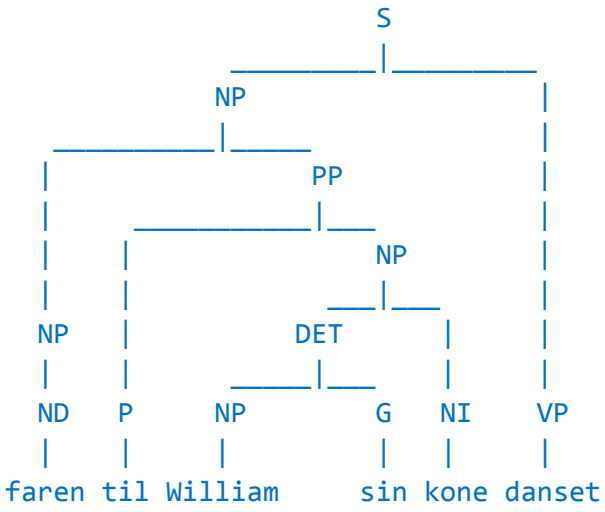
Oppgave 2 (samlet 15%)

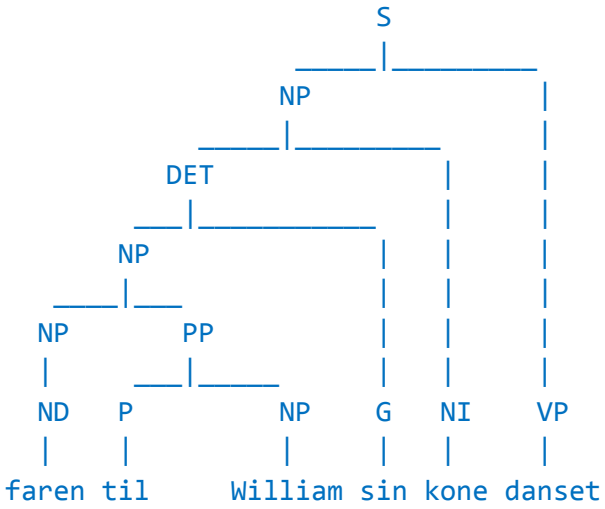
La grammatikk G være:

```
S   -> NP VP
NP  -> ND
NP  -> DET NI
NP  -> NP PP
PP  -> P NP
DET -> NP G
VP  -> lo | smilte | danset
ND  -> faren | mora | broren | søstera | kona | mannen
NI  -> far | mor | kone | mann | søster | bror
P   -> til
G   -> sin
DET -> en | ei
NP  -> William | Kate | Elizabeth | Philip |
       Charles | Michael | Pippa | Camilla
```

Spørsmål 2.1 (8%)
Tegn opp de to trærne grammatikken gir til ordsekvensen

3. *Faren til William sin kone danset*





Spørsmål 2.2 (7%)
Hvor mange forskjellige syntaktiske analyser har sekvens (4)?

4. *Faren til William sin kone sin søster danset*

Har alle de syntaktiske analysene ulik betydning? Forklar hvilke par som evt. har samme betydning og hvilke par som evt. har forskjellig betydning og hva forskjellene i betydning består i.

Setningen har tre syntaktiske analyser alle med forskjellig tolkning

```
A) (((Faren til William) sin kone) sin søster) danset
B) ((Faren til (William sin kone)) sin søster) danset
C) (Faren til ((William sin kone) sin søster)) danset
```

A: Faren til William er Charles. Hans kone er Camilla. Camilla sin søster danset.
B: William sin kone er Catherine. Hennes far er Michael. Michaels søster danset.
C: Williams kone er Catherine. Hennes søster er Pippa. Pippas far (= Michael) danset

(Det er ikke noe krav om å kjenne familierelasjonene mellom navnene. De er bare brukt for å gjøre sorteringen lettere).

Oppgave 3 (samlet 20%)

Spørsmål 3.2 (10%)
Vis hvordan en chart-parser for grammatikk G kan anerkjenne

1. *William sin kone danset*

```
|. William .   sin   .   kone  .  danset .|
|[---------]         .         .         .| [0:1] NP -> 'William' *
|.         [---------]         .         .| [1:2] G  -> 'sin' *
|.         .         [---------]         .| [2:3] NI -> 'kone' *
|.         .         .         [---------]| [3:4] VP -> 'danset' *
|>         .         .         .         .| [0:0] S  -> * NP VP
|>         .         .         .         .| [0:0] NP -> * NP PP
|>         .         .         .         .| [0:0] DET -> * NP G
|[--------->         .         .         .| [0:1] S  -> NP * VP
|[--------->         .         .         .| [0:1] NP -> NP * PP
|[--------->         .         .         .| [0:1] DET -> NP * G
|[-------------------]         .         .| [0:2] DET -> NP G *
|>         .         .         .         .| [0:0] NP -> * DET NI
|[------------------->         .         .| [0:2] NP -> DET * NI
|[---------------------------]         .| [0:3] NP -> DET NI *
|[---------------------------->         .| [0:3] S  -> NP * VP
|[---------------------------->         .| [0:3] NP -> NP * PP
|[---------------------------->         .| [0:3] DET -> NP * G
|[=====================================]| [0:4] S  -> NP VP *
(S (NP (DET (NP William) (G sin)) (NI kone)) (VP danset))
```

Chartet bør tegnes. Det bør også settes opp agenda. Kantene bør nummereres.

Spørsmål 3.1 (10%)
Omform grammatikk G til en grammatikk G1 på Chomsky-normalform (CNF) og vis hvordan CKY-algoritmen vil anerkjenne sekvens (4).

G1:
Legger til G:

```
NP  -> faren | mora | broren | søstera | kona | mannen
```

Fjerner:

```
NP  -> ND
```

```
In [170]: cky_table("faren til William sin kone sin søster danset".split(),
G1)
  (0)faren(1)til(2)William(3)sin(4)kone(5)sin(6)søster(7)danset(8)
```

0)	ND NP		NP	DET	NP	DET	NP	S
1)		P	PP		PP		PP	
2)			NP	DET	NP	DET	NP	S
3)				G				
4)					NI			
5)						G		
6)							NI	
7)								VP

Oppgave 4 (10%)

Er språket generert av G regulært? Begrunn svaret.

Ja språket kan beskrives ved en FSA:

```
START:
0

FINAL:
3

EDGES:
0  DET    1
0  PN     2
0  ND     2
1  NI     2
2 'til'   0
2 'sin'   1

ABRS:
VP: 'lo', 'smilte', 'danset'
ND: 'faren', 'mora', 'broren', 'søstera', 'kona', 'mannen'
NI: 'far', 'mor', 'kone', 'mann', 'søster', 'bror'
DET: 'en', 'ei'
PN: 'William', 'Kate', 'Elizabeth', 'Philip', 'Charles', 'Michael', 'Pippa',
'Camilla'
```

Oppgave 5 (20%)

Eiendomsrelasjoner og liknende relasjoner, f.eks. familierelasjoner, kan i norsk uttrykkes med *sin* som vi har sett så langt. Men de kan også uttrykkes med genitivs-s. For eksempel har setning (6) like mange analyser som setning (4) med tilsvarende betydninger.

2. *Faren til Williams kones søster danset*

Det er flere mulige måter å beskrive slike konstruksjoner. Her velger vi å betrakte *Williams* og de andre ordene med genitivs-s som vanlige ord og deretter endre de syntaktiske reglene i grammatikk G for å passe til dette. Først legger vi genetivsordene til leksikon.

```
ND  -> farens | moras | brorens | søsteras | konas | mannens
NI  -> fars | mors | kones | manns | søsters | brors
NP  -> Williams | Kates | Elizabeths | Philips |
       Charles' | Michaels | Pippas | Camillas
```

Du skal nå endre og utvide grammatikken med regler slik at den generer sekvens (6) med rett antall analyser. Genitivskonstruksjonen skal kunne gjentas vilkårlig mange ganger (*søsteras brors mors fars* ...). Du skal også utstyre grammatikkreglene med trekk slik at grammatikken ikke overgenerer. Spesielt skal den ikke generere

3. ** Moras danset*
4. ** Mora søster danset*

I tillegg skal du bruke trekk for å få riktig samsvar med den ubestemte artikkel:

ei mor, ei søster, ei kone
en mann, en far, en bror

Her skal vi ikke tillate

** ei mann, *ei far, *ei bror*

Men vi skal tillate

en mor, en søster, en kone

Deretter skal du tegne trærne – med trekk – som grammatikken gir til setningen:

5. *Faren til Williams kone danset*

```
S           -> NP[-POSS] VP
NP[POSS=?x] -> ND[POSS=?x]
NP[POSS=?x] -> DET[FEM=?y] NI[FEM=?y, POSS=?x]
NP[POSS=?x] -> NP[-POSS] PP[POSS=?x]
PP[POSS=?x] -> P NP[POSS=?x]
DET         -> NP[-POSS] G
VP          -> 'lo' | 'smilte' | 'danset'
ND[-POSS]   -> 'faren' | 'mora' | 'broren' | 'søstera' | 'kona' |
'mannen'
NI[-POSS, -FEM] -> 'far' | 'mann' | 'bror'
NI[-POSS]       -> 'mor' | 'kone' | 'søster'
P -> 'til'
G -> 'sin'
DET             -> 'en'
DET[+FEM]       -> 'ei'
NP[-POSS]       -> 'William' | 'Kate' | 'Elizabeth' | 'Philip' |
'Charles' | 'Michael' | 'Pippa' | 'Camilla'

ND[+POSS]       -> 'farens' | 'moras' | 'brorens' | 'søsteras' |
'konas' | 'mannens'
NI[+POSS, -FEM] -> 'fars' | 'manns' | 'brors'
NI[+POSS]       -> 'mors' | 'kones' | 'søsters'
NP[+POSS]       -> 'Williams' | 'Kates' | 'Elizabeths' | 'Philips' |
"Charles'" | 'Michaels' | 'Pippas' | 'Camillas'
DET             -> NP[+POSS]
```

```
(S[]
  (NP[-POSS]
    (NP[-POSS] (ND[-POSS] faren))
    (PP[-POSS]
      (P[] til)
      (NP[-POSS] (DET[] (NP[+POSS] Williams)) (NI[-POSS] kone))))
  (VP[] danset))

(S[]
  (NP[-POSS]
    (DET[]
      (NP[+POSS]
        (NP[-POSS] (ND[-POSS] faren))
        (PP[+POSS] (P[] til) (NP[+POSS] Williams))))
    (NI[-POSS] kone))
  (VP[] danset))
```

Oppgave 6 (samlet 20%)

Vi skal se på klassifisering og vi bruker navnesettet fra NLTK. Det inneholder 7944 navn som hver er tilordnet en klasse, *male* eller *female*. Vi legger til side 794 navn til testing og bruker 7150 navn til trening. Av disse 7150 navnene er 4500 klassifisert som *female* og 2650 klassifisert som *male*.

Spørsmål 6.1 (10%)

Vi vil lage en Naive Bayes-klassfikator. Denne ser bare på ett trekk, siste bokstav i navnet. For bokstaven *n* er det 335 navn i treningssettet som er merket som *female* og 425 som er merket som *male*. Forklar hvordan klassifikatoren vil gå frem for å klassifisere navnet *John*. Du trenger ikke glatte verdiene.

Naive Bayes sammenlikner sannsynligheten P(*female* | *John*) med P(*male* | *John*). Siden det enesete trekket vi bruker er siste bokstav blir dette det samme som å sammenlikne sannsynlighetene P(*female* | SISTE=*n*) med P(*male* | SISTE=*n*). Fra treningsmaterialet kan vi estimere disse til

$$\hat{P}(female\ |SISTE = n) = \frac{C(female, SISTE = n)}{C(SISTE = n)} = \frac{335}{335 + 425} = \frac{335}{760}$$

$$\hat{P}(male\ |SISTE = n) = \frac{C(male, SISTE = n)}{C(SISTE = n)} = \frac{425}{335 + 425} = \frac{425}{760}$$

altså vil den velge klassen *male*.

Siden det bare er ett trekk og vi kan beregne disse sannsynlighetene, så holder dette som løsning. Men det er heller ikke feil å bruke den generelle formen, der Naive Bayes bruker Bayes' formel og omformer som følger

$$P(female \mid John) = \frac{P(John \mid female)\, P(female)}{P(John)} = \frac{P(SISTE = n \mid female) P(female)}{P(John)}$$

og tilsvarende for *male*. Her får vi estimatene:

$$\hat{P}(female) = \frac{C(female)}{C(alle)} = \frac{4500}{7150}$$

$$\hat{P}(male) = \frac{C(male)}{C(alle)} = \frac{2650}{7150}$$

$$\hat{P}(SISTE = n) = \frac{C(SISTE = n)}{C(alle)} = \frac{760}{7150}$$

$$\hat{P}(SISTE = n \mid female) = \frac{C(female, SISTE = n)}{C(female)} = \frac{335}{4500}$$

$$\hat{P}(SISTE = n \mid male) = \frac{C(male, SISTE = n)}{C(female)} = \frac{425}{2650}$$

Setter vi inn i formelen, får vi

$$\hat{P}(female \mid John) = \frac{\hat{P}(SISTE = n \mid female)\hat{P}(female)}{\hat{P}(John)} = \frac{\frac{335}{4500} \times \frac{4500}{7150}}{\hat{P}(John)} = \frac{335}{7150} \times \frac{1}{\hat{P}(John)}$$

$$\hat{P}(male \mid John) = \frac{\hat{P}(SISTE = n \mid female) P(female)}{\hat{P}(John)} = \frac{\frac{425}{2650} \times \frac{2650}{7150}}{\hat{P}(John)} = \frac{425}{7150} \times \frac{1}{\hat{P}(John)}$$

Altså samme resultat som med den enkle utregningen.

Spørsmål 6.2 (5%)

Denne klassifikatoren vil måtte gi samme klasse til *John* og *Marilyn*. For å skille mellom dem tar vi derfor med et trekk til. Vi ser på suffiks av de to siste bokstavene. Det er 70 navn i klassen *female* som har *yn* som de to siste bokstavene, og 10 navn i klassen *male* som har *yn* som de to siste bokstavene.
Det er ingen *female*-navn som har *hn* som de to siste bokstavene, mens det er 3 *male*-navn som har det.
Hvordan vil denne klassifikatoren gå frem for å klassifiser *Marilyn* og *John*? Får noen av dem endret klasse når vi legger til dette trekket?

Endringen fra forrige oppgave er at

$$\hat{P}(female \mid John) = \frac{\hat{P}(John \mid female)\, \hat{P}(female)}{\hat{P}(John)} = \frac{\hat{P}(SISTE = n, SUF\ = hn \mid female)\hat{P}(female)}{\hat{P}(John)}$$

Den naive forutsetningen i NB antar at dette er det samme som

$$= \frac{\hat{P}(SISTE = n|female)\hat{P}(SUF = hn|female)\hat{P}(female)}{\hat{P}(John)}$$

Siden P(SUF = *hn* | *female*) = 0 blir dette uttrykket = 0, mens det tilsvarende uttrykket for *male* blir >0 og *John* får samme klasse som i forrige del.

For *Marilyn* får vi

$$\hat{P}(female|Marilyn) = \frac{\hat{P}(SISTE = n|female)\hat{P}(SUF = yn|female)\hat{P}(female)}{\hat{P}(Marilyn)}$$

$$= \frac{\frac{335}{4500} \times \frac{70}{4500} \times \frac{4500}{7150}}{\hat{P}(Marilyn)} = \frac{335 \times 70}{4500} \times \frac{1}{7150 \times \hat{P}(Marilyn)}$$

$$P(male|Marilyn) = \frac{\hat{P}(SISTE = n|male)\hat{P}(SUF = yn|male)\hat{P}(male)}{\hat{P}(Marilyn)}$$

$$= \frac{\frac{425}{2650} \times \frac{10}{2650} \times \frac{2650}{7150}}{\hat{P}(Marilyn)} = \frac{425 \times 10}{2650} \times \frac{1}{7150 \times \hat{P}(Marilyn)}$$

Siden $\frac{335 \times 70}{4500} > \frac{425 \times 10}{2650}$ blir *Marilyn* nå klassifisert som *female*.

Spørsmål 6.3 (5%)
Trekket *hn* opptrer svært sjelden. Det samme gjelder f.eks. for *wn* som forekommer 3 ganger i klassen *female* og en gang i klassen *male* og *ln* som forekommer en gang i klassen *male* og 0 ganger i klassen *female*. Forklar hvorfor dette kan bli et problem. En måte å rette opp dette på er å bruke glatting ("smoothing"), og den enkleste formen er å "legge-til-en" glatting ("add-one"). Hvordan vil du bruke denne metoden for å glatte verdiene på suffiks-trekket?

Hvis et trekk ikke er observert under trening for en klasse, så vil sannsynligheten for dette trekket gitt klassen bli 0. Hvis trekket opptrer med en observasjonen i testsettet, vil sannsynligheten for at denne observasjonen hører til denne klassen bli 0 uansett de andre trekkene. F.eks. hvis det ikke hadde vært noen navn som sluttet på *ln* i treningsmaterialet, men det hadde vært et slikt ord i testmaterialet (*Lincoln*), så ville vi ikke vite hvilken klasse vi skulle tilordne det. Men vi ville ønsket å kunne klassifisere det på grunnlag av at det slutter på *n*.

"Legg-til-en glatting": Vi vil anta at vi har sett alle verdier av trekket en gang mer enn vi har. For å få sannsynligheter må vi justere med antall mulige verdier for trekket. Siden det er 26 bokstaver i det engelske alfabet, vil det være 26*26=676 antall mulige verdier for suffikstrekket. De justerte sannsynlighetene vil da se ut som f,eks.

$$\hat{P}(SUF = wn\,|female) = \frac{3+1}{4500+676}$$

$$\hat{P}(SUF = wn\,|male) = \frac{1+1}{2650+676}$$

$$\hat{P}(SUF = ln\,|female) = \frac{0+1}{4500+676}$$

$$\hat{P}(SUF = ln\,|male) = \frac{1+1}{2650+676}$$

SLUTT