# Løsningsforslag eksamen 2025

## Vektorrom og klassifisering

*1a, 1b, 1d er flervalgsoppgaver som rettes automatisk.*

### 1c: Lengde-normalisering 2:

Dersom vi bruker lengde-normaliserte vektorer så blir cosinus-likhet det samme som prikkproduktet. Forklar hvorfor. Vis relevante formler som del av forklaringen din.

For to vektorer A og B er cosinus-likheten definert som (A · B) / (||A|| * ||B||),
der ||A|| tilsvarer lengden (den euklidske normen) til en vektor A. Dersom lengden for både A og B er 1 vil hele nevneren i brøken også bli 1 og dermed falle bort, slik at kun prikkproduktet gjenstår.

### 1e: Evaluering

Forklar kort forskjellen på ekstrinsisk og intrinsisk evaluering. Gi eksempler.

Ved intrinsisk evaluering tester vi kvaliteten på en modell på en måte som er direkte knyttet til oppgaven den er trent for. For spam-klassifikatorer kan vi f.eks. sammenlikne hvor godt de fungerer basert på F1-målet for hvor godt de klassifiserer eposter som spam og ikke-spam. Ved ekstrinsisk evaluering måler vi effekten mer indirekte, altså ved å måle effekten modellen har som del av en praktisk anvendelse. For spam-klassifikatorene våre kan vi f.eks. måle hvor fornøyde brukere er med epost-klienten sin gitt ulike versjoner av spam-filteret.

### 1f: Distribusjonell semantikk

I denne oppgaven skal du svare på tre relaterte spørsmål. Det bør holde med et par setninger for hvert punkt.

1. Forklar kort den distribusjonelle hypotesen.

Den distribusjonelle hypotesen sier at ord som forekommer i liknende kontekster også har liknende betydning. Gitt et tilstrekkelig stort korpus kan vi dermed måle hvor like to ord er semantisk sett, gjennom å sammenlikne hvilke kontekster de forekommer i.

2. Forklar kort hva som menes med henholdsvis antonymi og polysemi. Gi eksempler.

Antonymer er ord som har motsatt betydning av hverandre , f.eks. "liten" / "stor", "natt" / "dag". Altså det motsatte av synonymer.

Polysemer er ord som har flere ulike beslektede betydninger, f.eks. "fot" (i betydningen "kroppsdel" eller "nedre del", som i "fjellets fot").

3. Forklar kort hvordan disse to begrepene forholder seg til den distribusjonelle hypotesen.

Antonymer bryter tilsynelatende med grunnantakelsen i den distribusjonelle hypotesen, siden antonymer typisk vil forekomme nettopp i like kontekster, men likevel ha motsatt betydning. Samtidig er det mulig å argumentere for at betydningene likevel er like, siden de begge typisk beskriver ulike sider av samme underliggende fenomen; både "liten" og "stor" beskriver størrelse, både "dag" og "natt" beskriver tider av døgnet.
Om vi tar utgangspunkt i at den distribusjonelle hypotesen beskriver ordformer snarerer enn ordbetydninger så skaper polysemer og andre former for flertydighet problemer ved at ett og samme ord brukes i ulike kontekster avhengig av hvilken betydning av ordet som intenderes.

## Logistisk regresjon og sekvenser

### 2a: Logistisk regresjon 1

Du ønsker å utvikle et system for å automatisk klassifisere eposter i tre kategorier: **hast**, **ikke-hast**, og **spam**, kun basert på e-postemner. Systemet skal ta i bruk en multinomial logistisk regresjonsmodell med enkle *bag-of-words* trekk ekstrahert fra emnet.

Du har et treningssett som består av 8 epostemner:

| **Emne** | **Kategori** |
|---|---|
| haster ! feil i eksamen | hast |
| exclusive offer ! | spam |
| feil i artikkel | ikke-hast |
| neste møte | ikke-hast |
| reviews due today | hast |
| neste eksamen | ikke-hast |
| møte nå | hast |
| viagra offer | spam |

1) Hvor mange bag-of-words trekk vil modellen inneholde, kun basert på ordene som forekommer i treningssettet (tegnsetting inkludert) ?

Ordene/tegnsetting som forekommer i treningssettet er ['!', 'artikkel', 'due', 'eksamen', 'exclusive', 'feil', 'haster', 'i', 'møte', 'neste', 'nå', 'offer', 'reviews', 'today', 'viagra'], altså 15 bag-of-words trekk.

2) Hvor mange parametre (vekter og skjæringspunkter) vil den multinomiale logistiske regresjonsmodellen inneholde?

Vi har 15 vekter + 1 skjæringspunkt per klasse, og dermed (15+1)*3 = 48 parametre.

3) Finn verdier for modellparametrene slik at du får en perfekt klassifisering på treningssettet. Du trenger ikke gjøre noen kompliserte beregninger, det er nok om du tester ut ulike verdier inntil du finner verdier for parametrene som gir en feilfri klassifisering på treningsdata. Det er mange mulige løsninger.

La $w_{ord,klasse}$ være vekten knyttet til ordet «ord» for klassen «klasse» (enten «hast», «ikke hast», «spam»). Da kan vi f.eks. sette:

- $w_{haster,hast}$, $w_{reviews,hast}$ og $w_{nå,hast}$ til store positive verdier, f.eks. 100
- $w_{exclusive,spam}$, $w_{viagra,spam}$ til store verdier
- $w_{artikkel,ikke\text{-}hast}$, $w_{neste,ikke\text{-}hast}$ til store verdier
- Og alle andre parametre til null

Og det vil gi oss en feilfri klassifisering på treningsettet. Det er mange løsninger her.

4) Vil modellen fungere godt på nye eposter? Forklar i 2-3 setninger.

Nei, modellen vil trolig **ikke** fungere bra nye eposter. Først fordi emner som bruker andre ord som de 15 ordene vi har som bag-of-words vil få en vilkårlig klassifisering (siden de vil ha en trekkvektor med bare nullverdier). Og også fordi modellen er *overtrent*: det er ikke slik at eposter med f.eks. «artikkel» eller «neste» i emnet alltid tilhører klassen «ikke-hast», selv om det er slik i treningssettet. Med andre ord vil modellen har en dårlig *generaliseringsevne*.

## 2b: Logistisk regresjon 2

1) Hva er **læringsraten** (eng. *learning rate*), og hva er dens rolle i treningen av maskinlæringsmodeller som logistisk regresjon ?

Læringsraten er en viktig hyperparameter i treningen av maskinlæringsmodeller, inkludert logistisk regresjon. Dens rolle er å styre hvor raskt modellen tilpasser seg data under trening ved å bestemme størrelsen på skrittene som tas i retning av den negative gradienten av kostnadsfunksjonen under gradientnedstigning (eng. gradient descent). Det er viktig å velge en passende læringsrate for å balansere mellom konvergenshastighet og nøyaktighet i modellens tilpasning til treningsdataene.

2) Hva tror du vil skje om vi skulle ta i bruk en *negativ læringsrate* til å trene en maskinlæringsmodell ? Forklar, basert på hvordan Gradient Descent fungerer.

Normalt, når læringsraten er positiv, beveger modellen seg mot minimumet ved å redusere verdien av kostnadsfunksjonen. Men hvis læringsraten er negativ, vil modellen i stedet bevege seg i motsatt retning. Dette ville føre til at kostnadsfunksjonen *øker* i stedet for å minske, og modellen vil divergere i stedet for å konvergere mot et minimum. I praksis ville dette resultere i

en modell som blir stadig dårligere til å tilpasse seg treningsdataene, og som til slutt blir ubrukelig for prediksjon eller klassifisering.

## 2c: Sekvensmerking 1

Du ønsker å bruke en HMM (*Hidden Markov Model*) til å trene en modell for å gjenkjenne navngitte entiteter (*named entity recognition,* eller NER), og har samlet et lite korpus med 3 setninger merket med entiteter.

1. Norge (*STED*) har mange byer.
2. Oslo (*STED*) og Mo i Rana (*STED*) er norske byer.
3. Det er mange byer i Europa (*STED*).

1. Del hver setning i tokens, og bruke en BIO-markering til å koble hver token til et merkelapp. Husk tegnsetting!

(Norge, B-STED), (har, O), (mange, O), (byer, O), (., O)

(Oslo, B-STED), (og, O), (Mo, B-STED), (i, I-STED), (Rana, I-STED), (er, O), (norske, O), (byer, O), (., O)

(Det, O), (er, O), (mange, O), (byer, O), (i, O), (Europa, B-STED), (., O)

2. Estimer transisjonsmodellen (uten smoothing) basert på disse 3 setningene, og skriv den i en tabell. Inkluder startsymbolet <s> i tabellen.

Hvis studenten har tatt med «.»:

| | O | B-STED | I-STED |
|---|---|---|---|
| <s> | 1/3 | 2/3 | 0 |
| O | 10 /12 | 2/12 | 0 |
| B-STED | 3/4 | 0 | 1/4 |
| I-STED | 1/2 | 0 | 1/2 |

Og hvis ikke:

| | O | B-STED | I-STED |
|---|---|---|---|
| <s> | 1/3 | 2/3 | 0 |

| | | | |
|---|---|---|---|
| O | 8/10 | 2/10 | 0 |
| B-STED | 2/3 | 0 | 1/3 |
| I-STED | 1/2 | 0 | 1/2 |

## 2d: Sekvensmerking 2

En værstasjon bruker en Hidden Markov Modell (HMM) for å finne den mest sannsynlige sekvensen av værforhold over flere dager basert på avlesninger fra en fuktighetssensor. Selve værforholdet er altså skjult, mens fuktigheten kan observeres.

De tre mulige tilstandene for værforholdet er enten sol (**S**), overskyet (**O**) eller regn (**R**). Fuktighetssensoren har tre mulige verdier: lav (**L**), middels (**M**) eller høy (**H**).

Vi bruker følgende *transisjonsmodell* som beskriver utviklingen i værforholdet fra dag til dag:

| | | *vær på dag i:* | | |
|---|---|---|---|---|
| | | sol (S) | overskyet (O) | regn (R) |
| *vær på dag i-1:* | sol (S) | 0.5 | 0.4 | 0.1 |
| | overskyet (O) | 0.3 | 0.5 | 0.2 |
| | regn (R) | 0.2 | 0.5 | 0.3 |

Og vi bruker også følgende *emisjonsmodell* som beskriver forholdet mellom vær og observert fuktighet på samme dag:

| | | *observert fuktighet på dag i:* | | |
|---|---|---|---|---|
| | | lav (L) | middels (M) | høy (H) |
| *vær på dag i:* | sol (S) | 0.4 | 0.5 | 0.1 |
| | overskyet (O) | 0.2 | 0.5 | 0.3 |
| | regn (R) | 0.0 | 0.2 | 0.8 |

La oss nå anta at værstasjonen mottar følgende fuktighetsavlesninger for tre dager på rad: [**middels**, **høy**, **middels**]

Bruk *beam search*-dekoding med en beam-bredde på 2 for å finne den mest sannsynlige sekvensen av værforhold over de tre dagene. Vis dine beregninger. For den første dagen i sekvensen kan du anta en uniform startfordeling (det vil si at sol, overskyet og regn alle har en sannsynlighet = 0,33).

**Dag 1:**

Her har vi kun tre muligheter:

$P(fukt_1 = M \mid dag_1 = S)\ P(dag_1 = S \mid \langle start \rangle) = 0.5 \times 0.33$

$P(fukt_1 = M \mid dag_1 = O)\ P(dag_1 = O \mid \langle start \rangle) = 0.5 \times 0.33$

$P(fukt_1 = M \mid dag_1 = R)\ P(dag_1 = R \mid \langle start \rangle) = 0.2 \times 0.33$

Med en beam-size på 2 beholder vi de to beste hypotesene, nemlig: {[S]=0.165, [O]=0.165}

**Dag 2:**

Her må vi vurdere 6 muligheter:

$P(fukt_2 = H \mid dag_2 = S)\ P(dag_2 = S \mid dag_1 = S)\ P([S]) = 0.1 \times 0.5 \times 0.165$

$P(fukt_2 = H \mid dag_2 = O)\ P(dag_2 = O \mid dag_1 = S)\ P([S]) = 0.3 \times 0.4 \times 0.165$

$P(fukt_2 = H \mid dag_2 = R)\ P(dag_2 = R \mid dag_1 = S)\ P([S]) = 0.8 \times 0.1 \times 0.165$

$P(fukt_2 = H \mid dag_2 = S)\ P(dag_2 = S \mid dag_1 = O)\ P([O]) = 0.1 \times 0.3 \times 0.165$

$P(fukt_2 = H \mid dag_2 = O)\ P(dag_2 = O \mid dag_1 = O)\ P([O]) = 0.3 \times 0.5 \times 0.165$

$P(fukt_2 = H \mid dag_2 = R)\ P(dag_2 = R \mid dag_1 = O)\ P([O]) = 0.8 \times 0.2 \times 0.165$

De to beste hypotesene fram til dag 2 er altså {[O,O]=0.02475, [O,R]=0.0264}

**Dag 3**

$P(fukt_3 = M \mid dag_3 = S)\ P(dag_3 = S \mid dag_2 = O)\ P([O,O]) = 0.5 \times 0.3 \times 0.02575$

$P(fukt_3 = M \mid dag_3 = O)\ P(dag_3 = O \mid dag_2 = O)\ P([O,O]) = 0.5 \times 0.5 \times 0.02575$

$P(fukt_3 = M \mid dag_3 = R)\ P(dag_3 = R \mid dag_2 = O)\ P([O,O]) = 0.2 \times 0.2 \times 0.02575$

$P(fukt_3 = M \mid dag_3 = S)\ P(dag_3 = S \mid dag_2 = R)\ P([O,R]) = 0.5 \times 0.2 \times 0.0264$

$P(fukt_3 = M \mid dag_3 = O)\ P(dag_3 = O \mid dag_2 = R)\ P([O,R]) = 0.5 \times 0.5 \times 0.0264$

$P(fukt_3 = M \mid dag_3 = R)\ P(dag_3 = R \mid dag_2 = R)\ P([O,R]) = 0.2 \times 0.3 \times 0.0264$

De to beste hypotesene fram til dag 3 er altså {[O,O,O]=0.0061875, [O,R,O] = 0.0066}

Den beste hypotesen er altså [overskyet,regn,overskyet].

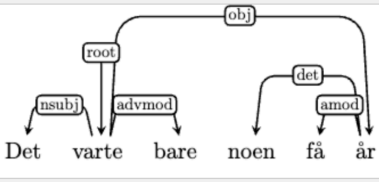
## Dependenssyntaks og parsing

### Oppgave 3a-d: Dependenssyntaks

*Flervalgsoppgaver, rettes automatisk.*

### Oppgave 3e: Dependensparsing

**3(e)** **Dependensparsing**




Ta utgangspunkt i vedlagte graf og angi en fullstendig transisjonssekvens som vil resultere i denne grafen ved bruk av "arc standard"-algoritmen for dependensparsing. Denne har tre mulige transisjoner: SHIFT , LEFT-ARC , og RIGHT-ARC.

Du skal for hver transisjon angi:

- innholdet i stack
- innholdet i inputbuffer
- valgt transisjon
- eventuell relasjon som legges til grafen

Merk at du skal starte og slutte med kun rotnoden på stacken. Når vi starter har inputbufferen denne setningen:

*Det varte bare noen få år*

| Stack | Buffer | Transisjon | Dependensrelasjon |
|---|---|---|---|
| [root] | [Det, varte, bare, noen, få, år] | ? | |
| ... | ... | ... | ... |

**Skriv ditt svar her** Hjelp

**Løsningsforslag:**

| Stack | Buffer | Transisjon | Dependensrelasjon |
|---|---|---|---|
| [root] | [Det, varte, bare, noen, få, år] | SHIFT | |
| [root, Det] | [varte, bare, noen, få, år] | SHIFT | |

| | | | |
|---|---|---|---|
| [root, Det, varte] | [bare, noen, få, år] | LEFT-ARC_nsubj | varte -> Det |
| [root, varte] | [bare, noen, få, år] | SHIFT | |
| [root, varte, bare] | [noen, få, år] | RIGHT-ARC_advmod | varte -> bare |
| [root, varte] | [noen, få, år] | SHIFT | |
| [root, varte, noen] | [få, år] | SHIFT | |
| [root, varte, noen, få] | [år] | SHIFT | |
| [root, varte, noen, få, år] | [] | LEFT-ARC_amod | år -> få |
| [root, varte, noen, år] | [] | LEFT-ARC_det | år -> noen |
| [root, varte, år] | [] | RIGHT-ARC_obj | varte -> år |
| [root, varte] | [] | RIGHT-ARC_root | root -> varte |
| [root] | [] | | |

Oppgave 3f: dependensparsing

### Dependensparsing 2

I kurset har vi sett på et alternativ til "arc standard"-algoritmen, nemlig "arc eager"-algoritme. Her skal du illustrere forskjellen(e) mellom disse ved å vise de første fire transisjonene fra forrige oppgave ved bruk av "arc eager" i stedet for "arc standard"-algoritmen.

(**NB!** merk at ved bruk av arc eager har vi ikke krav om at rotnoden befinner seg på stacken til slutt).

**Løsningsforslag:**

(Ved arc eager har vi følgende forskjeller:

- transisjoner gjennomføres mellom elementene på toppen av stacken og foran i bufferen (og ikke de to øverste elementene på Stack'en)
- LEFT-ARC fører til at dependenten fjernes fra Stack'en etter applikasjon
- RIGHT-ARC fører til at dependenten flyttes til Stack'en
- REDUCE ny relasjon som fjerner det øverste elementet på Stack'en)

| Stack | Buffer | Transisjon | Dependensrelasjon |
|---|---|---|---|

| | | | |
|---|---|---|---|
| [root] | [Det, varte, bare, noen, få, år] | SHIFT | |
| [root, Det] | [varte, bare, noen, få, år] | LEFT_ARC_nsubj | varte -> Det |
| [root] | [varte, bare, noen, få, år] | RIGHT-ARC_root | root -> varte |
| [root, varte] | [bare, noen, få, år] | RIGHT-ARC_advmod | varte -> bare |

## Maskinoversettelse og dialogsystemer

### 4a: Maskinoversettelse

Hvorfor er det problematisk å bruke BLEU til evaluering av maskinoversettelsemodeller når målspråket har en fri ordstilling? Begrunn svaret ditt basert på BLEU-formelen.

BLEU fungerer ved å telle hvor mange *n*-grammer (sekvenser av *n* ord) i den maskinoversatte teksten som også finnes i en eller flere referansetekster. Presisjonen beregnes som forholdet mellom antall n-grammer i maskinoversettelsen som matcher med n-grammer i referanseteksten, delt på totalt antall n-grammer i maskinoversettelsen. I språk med fri ordstilling kan mange ord bytte plass samtidig som setningen fortsatt er både grammatisk og semantisk korrekt. Dette vil kunne lede til lavere BLEU-scorer, selv om oversettelsen er god.

### 4b: Dialogsystemer 1

1) Hva er **retrieval-augmented generation** (RAG)? Forklar med 2-3 setninger.

Retrieval-augmented generation (RAG) er en metode som kombinerer en språkmodell med et søkesystem som skal hente relevant informasjon fra en tekstdatabase. Når modellen mottar en forespørsel/spørsmål, henter den først relevante tekstbiter fra en ekstern kunnskapsdatabase (often en samling av tekstdokumenter), og kombinerer deretter disse tekstbitene med forespørselen for å generere svaret. Dette skjer i to trinn: først henting (retrieval), deretter generering (generation).

2) Nevn minst 2 fordeler med RAG-baserte løsninger sammenlignet med språkmodeller uten retrieval-komponent.

- Oppdatert informasjon: RAG kan hente inn fersk og relevant informasjon fra eksterne kilder, noe som gjør svarene mer aktuelle og nøyaktige.
- Mulighet til å fjerne utdaterte eller feilaktige informasjon ved å ta bort de tilsvarende dokumentene uten å måtte trene modellen på nytt
- Redusert hallusinasjon: Ved å støtte seg på faktiske dokumenter reduseres risikoen for at modellen "finner på" informasjon.

- Mulighet til å ta i bruk interne data (f.eks. en intranet eller bedriftsdatabase) som språkmodellen ikke har “sett” under treningen, fordi den ikke var åpent tilgjengelig
- Eksplisitt skille mellom “språkferdigheter” og “faktakunnskap”
- Gjør det lettere å forklare hvorfor modellen gir et bestemt svar (ved å inspirere tekstbitene som ble hentet ut av søkesystemet)

## 4c: Dialogsystemer 2

Kan IR-baserte chatboter føre til etiske utfordringer? Hvis ja, forklar hvilke problemer kan oppstå.

Ja, det er mange etiske utfordringer som kan oppstå når man utvikler slike chatboter. Det viktigste er kanskje at IR-baserte chatboter henter sine svar direkte fra korpuset. Diskriminerende eller hatefulle ytringer som befinner seg i korpuset risikerer altså å gjentas i samtale med brukere. Det er også en viss risiko for at chatboten vil gi feilaktige svar til brukeren, enten fordi ytringen som ble valgt inneholder ukorrekte opplysninger, eller fordi ytringen ikke samsvarer godt med spørsmålet som ble stilt.