IN1140 H2017 – Oblig 3a
Betinget sannsynlighet og HMM-tagging

Innleveringsfrist: 05.11 kl. 23.59

Lever inn svarene dine i Devilry (`https://devilry.ifi.uio.no/`) i filer som angir brukernavnet ditt, slik: `oblig3a_brukernavn.py`. Pass på at filen din kan kjøres som et program.

En perfekt besvarelse på denne oppgaven er verdt 100 poeng. Minner om at for å bestå kreves minst 100 poeng (av 200 mulige) for hver av oblig 2(a+b) og 3(a+b).

1 Betinget sannsynlighet (50 poeng)

I denne oppgaven ser vi på hvordan vi kan beregne sannsynlighetene som er byggesteinene i en HMM tagger, dvs. **observasjonssannsynligheter** (*emission probabilities*) $P(w_i|t_i)$ og **transisjonssannsynligheter** (*transition probabilities*) $P(t_i|t_{i-1})$, og hvordan vi lett kan beregne disse med innebygd funksjonalitet i NLTK.

Her går vi gjennom seks trinn for å forklare hvordan oppgaven skal løses. Merk at det ikke er spørsmål i hvert trinn.

1. For å beregne frekvens av (tagg, ord) par for hele Brown-korpuset kan vi bruke `nltk.ConditionalFreqDist()` som følgende:

   ```
   >>> brown_ord_tagger = brown.tagged_words()
   >>> brown_tagger_ord =
   [(tag, word) for (word, tag) in brown_ord_tagger]
   >>> cfd_tagger_ord = nltk.ConditionalFreqDist(brown_tagger_ord)
   ```

   Slik at `brown_tagger_ord = [(tag, word) for (word, tag) in brown_ord_tagger]` er en *list comprehension* som gir samme resultat som å skrive:

   ```
   >>> brown_tagger_ord = []
   >>> for (word, tag) in brown_ord_tagger:
           brown_tagger_ord.append((tag, word))
   ```

   For å forstå hvordan `nltk.ConditionalFreqDist()` funksjonen fungerer kan du lese kapittel fire og fem i NLTK-boken. Frekvensdistribusjonen har ordklassetaggene som betingelser, dvs. at om vi kjører `cfd_tagger_ord.conditions()` får vi en liste med ordklassetagger (feks. NN, DT, samt en rekke andre tagger) for distribusjonen `cfd_tagger_ord`. Om du har lyst å vite litt mer om list comprehensions se `https://docs.python.org/2/tutorial/datastructures.html`.

Du kan bruke `most_common()` for å finne ut det mest frekvente ordet av en gitt ordklasse (feks. `TAG`) som følgende:

```
>>> mest_frekvente = cfd_tagger_ord['TAG'].most_common()[0]
```

For å kunne finne ut hvor ofte et ord (feks. `ord`) forekommer som en gitt ordklasse (feks. `TAG`), kan vi gjøre følgende:

```
>>> tag_ord = cfd_tagger_ord['TAG']['ord']
```

2. Besvar følgende spørsmål ved hjelp av frekvensdistribusjonen `cfd_tagger_ord` og eksemplene over:

   (a) Hva er det mest frekvente substantivet?

   (b) Hvor ofte forekommer substantivet *linguist*?

   (c) Hva er det mest frekvente adjektivet?

   Her skal du altså skrive Python-kode som beregner og skriver ut svaret på oppgavene (a)-(c) over.

3. Vi gjør om frekvensdistribusjonen (her med variabelnavnet `cfd_tagger_ord`) til en sannsynlighetsdistribusjon (basert på Maximum Likelihood Estimation (MLE)) med funksjonen `nltk.ConditionalProbDist()`, slik:

   ```
   cpd = nltk.ConditionalProbDist(cfd_tagger_ord, nltk.MLEProbDist)
   ```

   Om du kjører `cpd['JJ'].max()` vil det gi deg det mest sannsynlige adjektivet,
   og `cpd['JJ'].prob("new")` gir deg sannsynligheten for *new*, gitt en adjektivtagg (altså P(*new*|JJ)).

4. For å beregne transisjonssannsynlighetene trenger vi frekvensinformasjon for bigrammer av tagger. Her bruker vi `nltk.bigrams()` som tar inn en liste og gir ut alle bigrammer basert på denne:

   ```
   >>> liste = ["the","man","saw","her"]
   >>> nltk.bigrams(liste)
   [('the', 'man'), ('man', 'saw'), ('saw', 'her')]
   ```

   I motsetning til eksempelet over skal du her trekke ut bigrammer av tagger (og ikke ord):

   ```
   >>> liste = ["DT","NN","VBD","PP$"]
   >>> nltk.bigrams(liste)
   [('DT', 'NN'), ('NN', 'VBD'), ('VBD', 'PP$')]
   ```

   Trekk ut alle bigrammer av tagger fra Brown-korpuset med `nltk.bigrams()` og lagre disse i en variabel. For å få en liste av tagger, må du gjøre følgende:

   ```
   >>> brown_tagger = [tag for (word, tag) in brown_ord_tagger]
   ```

5. Bruk `nltk.ConditionalFreqDist()` til å beregne frekvens for bigrammene av tagger og besvar følgende spørsmål (her skal du altså skrive Python-kode som beregner og skriver ut svaret på oppgavene (a)-(b) under):

   (a) Hvilken tagg er det som oftest etterfølger et substantiv i Brown-korpuset?

   (b) Hvor ofte forekommer bigrammet 'DT NN'?

6. Igjen kan du utlede en sannsynlighetsdistribusjon fra frekvenstellingene over bigrammene med `nltk.ConditionalProbDist()` (som i 1.3).

   (a) Hva er den betingede sannsynligheten for et substantiv, gitt et determinativ (altså P(NN|DT))? Svaret skal også her beregnes og printes fra Python-skriptet ditt.

Merk:

- Benytt deg av eksemplene som ble gitt utover oppgave 1 for å løse oppgaven.
- Les nøye gjennom hele oppgaven, da samme variabelnavn er brukt flere steder.

2 HMM-tagging (50 poeng)

I denne oppgaven skal vi sammenligne sannsynligheten for to taggsekvenser for samme setning. Ta en titt på J&M kapittel 5.5.1, side 176-178 og gå fram på nøyaktig samme måte for å løse denne oppgaven.

Her er setningen med to taggsekvenser:

(a)	PPSS	VBD	PP$	NN
	I	saw	her	duck

(b)	PPSS	VBD	PPO	VB
	I	saw	her	duck

(PPSS er taggen for personlige pronomen i nominativ som ikke er 3. person entall, feks. *I, they*. PP$ er taggen for possessive pronomen feks. *my, your*, og PPO er taggen for personlige pronomen i akkusativ feks. *him, you*).

Som i eksempelet fra J&M starter setningene på samme måte og taggsekvensene har kun fire sannsynligheter som skiller seg fra hverandre. Vi ser her bort fra flertydigheten for *saw*.

Bruk koden fra Oppgave 1 og hele Brown-korpuset for å beregne transisjonssannsynligheter, f.eks. P(PP$|VBD), og observasjonssannsynligheter, f.eks. P(*her*|PP$). Bruk disse sannsynlighetene til å beregne sannsynligheten for den delen av taggsekvensen der (a) og (b) er forskjellig (dvs de to ulike sekvensene for *her duck*).[1]

Hvilken lesning er mest sannsynlig?[2] Her skal du beregne sannsynlighetene for begge (med Python-kode), og vise hvilken som er mest sannsynlig.

[1] **Hint 1**: Her skal du bruke `cpd_tagger_ord['TAG'].prob("ord")` for å beregne observasjonssannsynligheter, og din sannsynlighetsdistribusjon over taggbigram (altså din løsning på oppgave 1.6) for å beregne transisjonssannsynligheter.

[2] **Hint 2**: Dette skal løses slik det er gjort i J&M kapittel 5.5.1.