INF1820 V2015 — Oppgave 2b
Sannsynlighet og tagging

Innleveringsfrist, onsdag 15. april

Lever inn svarene dine i Devilry (https://devilry.ifi.uio.no) en fil som angir brukernavnet ditt, slik: oblig2b_brukernavn.py

En perfekt besvarelse på denne oppgaven er verdt 100 poeng.

1 Tilfeldig samsvar i annotering (15 poeng)

Arild og Birgitte annoterer et korpus med orklasser i et datalingvistisk forskningsprosjekt. Dessverre velger de ordklasser tilfeldig i stedet for å lese teksten, fordelt slik:

- I 20% av tilfellene velger de taggen det
- I 55% av tilfellene velger de taggen subst
- I 25% av tilfellene velger de taggen verb

Det vil si, for et gitt ord er det 55% sjanse for at Arild velger subst og 55% sjanse for at Birgitte velger subst. Hva er sjansen for at Arild og Birgitte velger samme tagg for et ord?

I denne oppgaven kan du godt bruke Python, men behøver det ikke. Det holder å vise utregningene. Begrunn svaret ditt.

2 Zipfianske distribusjoner (35 poeng)

Hvis frekvensene til ord i et korpus har en Zipfiansk distribusjon, er det slik at hvis ordene rangeres etter frekvens vil det andre (nest mest frekvente) ordet forekomme halvparten så ofte som det første, det tredje ordet en

tredjedel så ofte, osv. Altså er ordets frekvens (f) omvendt proporsjonal med rangen (r):

$$f = \frac{k}{r}$$

der k er en konstant.

Hvis vi tar utgangspunkt i Brown-korpuset, husker vi at det mest frekvente ordet (det vil si rang 1), *the*, forekom omlag 70.000 ganger, og det neste ordet (*of*) forekom ca. 36.000 ganger, nesten akkurat det Zipfs lov forutsier.

Vi sier at (det ikke-eksisterende) IFI-korpuset inneholder 1 million ord og 56.000 distinkte ord (altså *typer*, ikke tokens), der det mest frekvente ordet forekommer 63.287 ganger. Bruk Python til å regne ut frekvensen for hver av de 56.000 typene i korpuset, under antagelsen at de er fordelt nøyaktig slik Zipfs lov forutsier.

For å løse denne oppgaven trenger du litt ekstra info om Python og desimaltall. Det er slik at hvis begge tallene i en deleoperasjon (/) er heltall, vil også resultatet være heltall (det største heltallet som er mindre en svaret med desimaler, for å være helt presis), og 2/3 vil returnere 0. For å få desimaltall må heltall konverteres til flyttall med funksjonen round(): float(2)/float(3) returnerer 0.66666.

Til slutt må vi runde av tall, siden uttrykket vi bruker til å beregne frekvens sjelden vil returnere heltall. Bruk den innebygde funksjonen round() til dette: round(float(2)/float(3)) returnerer 1.0.

Hvor mange ord har frekvens 1?

3 Flertydighet og ordklassetagger (15 poeng)

Velg tre flertydige overskrifter fra siden http://fun-with-words.com/ambiguous_headlines.html og for hver av dem, gjør følgende:

- Formuler tvetydigheten med egne ord
- Tagg overskriften med ordklassetagger fra listen under. Bruk formatet ord/tagg: Book/V that/DT flight/NN
- Angi om kunnskap om ordklasser entydiggjør overskriften

Bruk følgende taggsett:

DT	determinativ/bestemmer (*the*, *a*)
NN	fellesnavn (*table*, *dog*)
NNP	egennavn (*Astrid*, *Oslo*)
JJ	adjektiv (*red*, *serious*)
V	verb (*dance*, *love*)
CC	konjunksjon (*and*, *or*)
P	preposisjon (*of*, *on*)
RB	adverb (*slowly*, *tomorrow*)

Hvis du er usikker, kan du konsultere manualen som ble skrevet for annoteringen av Penn Treebank: ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz

4 Tagging med regulære uttrykk (35 poeng)

I denne oppgaven skal du lage en ordklassetagger med regulære uttrykk.

Taggeren med regulære uttrykk i NLTK-boka (del 5.4, under overskriften *The Regular Expression Tagger*) sjekker kun noen få uttrykk, så her er det masse rom for forbedring: for eksempel kan vi tagge alle ord på *–able* som adjektiv. Definer en tagger ved hjelp av nltk.RegexpTagger der du har minst 10 uttrykk i tillegg til de som er nevnt i boka. Dokumenter alle reglene dine, og gi minst ett eksempel på ord som dekkes.

Husk å håndtere liten og stor bokstav, og at enkeltord også kan brukes i de regulære uttrykkene, slik at f.eks. *the* alltid vil tagges som bestemmer.

Bruk adventure-kategorien i Brown mens du utvikler (altså skriver regulære uttrykk) taggeren din ved å teste å teste nøyaktigheten (en: *accuracy*) til taggeren. Når du er fornøyd med reglene dine, test taggeren på kategorien fiction i Brown, og rapporter resultatene. Det er viktig at du ikke endrer reglene dine etter dette.

Merk: Poengene du får på denne oppgaven er ikke basert på nøyaktigheten til taggeren i den endelige evalueringen på fiction-kategorien! Det er også helt vanlig at nøyaktigheten til en tagger synker når man tester den på et annet korpus enn den ble utviklet.

Til slutt skal programmet ditt lese inn filen test_setninger.txt som er lagt ut sammen med denne obligen, tagge den, og skrive ut resultatet. Kopier outputen inn i filen din og diskuter minst 3 av feilene taggeren gjør, og kom med forslag til hvordan den kan forbedres.