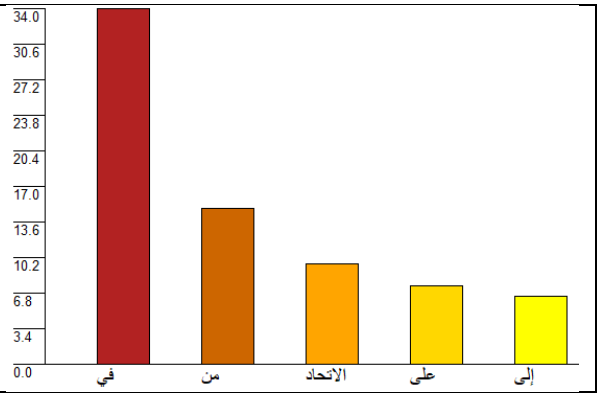
Visualisering av de fem mest frekvente ordene i en tekst

Denne oppgaven har to deler. Den første delen går på selve analysen, mens den andre delen går på å visualisere resultatene fra del 1. Når du er ferdig med alle delene, kan du prøve å utvide oppgaven til å inkludere flere enn de fem første ordene.



Del 1

Oppgave 1
Finn én eller flere tekster som du setter sammen. Dette er materialet du skal bruke. Lagre hele teksten i en variabel som du kan bruke i programmet. Python håndterer veldig mange språk godt, så her trenger du ikke begrense deg til norsk eller engelsk, men språket bør bruke mellomrom for å gjøre de neste stegene lettere.

Oppgave 2

Hvis du vil bruke nltk:
Last ned NLTK
http://www.nltk.org/install.html
og bruk funksjonen word_tokenize() på tekststrengen.

Hvis ikke:
Hvis du ikke vil bruke NLTK, kan du prøve å tenke på hvordan du kan tokenisere teksten selv, og lage en prosedyre som gjør det. Bruk .split() for å dele opp teksten.

Oppgave 3
Lag en funksjon som rensker uønskede "ord" fra ei liste, og returnerer den renskede lista. Tegn som ", ? og ! er i mange tilfeller ikke like interessante, men ender ofte opp som ganske frekvente. Disse kan vi fjerne før vi teller ordene.

Oppgave 4
Lag en funksjon som teller forekomstene av alle ordene i en tekst og legger dem i ei ordbok. Funksjonen skal returnere den ferdige ordboka. Ordboka kan ha formen "ord":antall.

Oppgave 5
Lag en funksjon som gjør om ordboka til ei nøsta liste som kan sorteres, og returnerer den ferdigsortert liste.

Del 2
Nå skal du lage en grafisk representasjon av dataene. Bruk ezgraphics. Søylene kan lages av firkanter, mens linjene er streker. La den første søylen (tekstens aller mest frekvente

ord) definere makshøyden, og la den ha en y-verdi som er et tall *avstandFraToppen* fra toppen. Da kan de andre søylene ha startpunkt med en y-verdi som tilsvarer:

søyle$_i$ * round((frekSøyle$_i$ *(HØYDE / frekSøyle1))-avstandFraToppen)

TIPS

Oppgave 2

Mange av de "problematiske" tegnene vil komme enten aller først eller aller sist i et ord, du kan derfor sjekke om ordene i teksten om noen av tegnene forekommer der.

Oppgave 3

Opprett ei ny liste som skal ta vare på alle ord som ikke er tegn.

Oppgave 4

Hvis et ord ikke finnes i ordboka, kan det legges til med frekvens 1. Hvis det allerede finnes, kan det legges til, og vi kan øke frekvensen med 1.

Oppgave 5

For å kunne sortere lista ordentlig, bør den nøsta lista ha frekvensene som første element i hvert par, altså på formen [[frekvens,"ord"],[frekvens,"ord"],...]. Da kan man bruke funksjonen sorted(), og gi den argumentet *reverse=True*. Altså: sortertListe = sorted(liste, reverse=True). Uten reverse=True havner det mest frekvente ordet bakerst.

Noen forhold å ta hensyn til.