IN1140 Obligatorisk innlevering 1a: Morfologi, lingvistiske nivåer, og tekst i Python

Høst 2025

I denne oppgaven skal vi se litt på lingvistiske nivåer og morfologi. Vi skal også jobbe med tekst i Python. Vi må derfor kunne lese inn filer og jobbe med tekststrenger. Vi skal også gjøre et første forsøk på et program som utfører såkalt tokenisering, dvs. bryter en tekst opp i ord. Tokenisering er et nødvendig utgangspunkt for de aller fleste språkteknologiske oppgaver.

Innleveringsfrist: 15.09 kl. 23.59

Innlevering av oppgaven skjer i Devilry. Det er viktig at du setter deg inn i retningslinjene for obligatoriske oppgaver. Besvar oppgavene i én fil som angir brukernavnet ditt slik:

`oblig1a_brukernavn.py`

Du skal også levere inn filen `in01.txt` slik at retter kan reprodusere tokeniseringskoden din. Ikke pakk filene inn i en ZIP-mappe. Besvarelsen må kunne kjøres i Python fra terminalen med `python oblig1a_brukernavn.py`, eventuelt `python3 oblig1a_brukernavn.py`.

Poeng En perfekt løsning av denne innleveringen er verdt 100 poeng. Koden må kunne kjøres og må inneholde kommentarer som forklarer hva koden gjør. Løsningen på oppgave 1 og 2 skal skrives som kommentarer i Python-filen din.

1 Lingvistiske nivåer (eksamen H2017) (10 poeng)

Hvilke nivåer av analyse opererer vi med i lingvistikk? Forklar også helt kort hva som er fokuset for analyse på hvert nivå. En setning om hvert nivå er nok.

2 Morfologi (eksamen V2012) (10 poeng)

a) Morfemer kan klassifiseres som enten bundne eller frie. Hva er forskjellen på bundne og frie morfemer? Gi minst 3 eksempler på frie morfemer.

b) De bundne morfemene (affikser) kan videre deles inn i minst to undergrupper (Ulike språk opererer med ulike typer bundne morfemer). Hva er disse undergruppene, og hva skiller dem fra hverandre? Gi noen eksempler.

3 Strenger i Python (30 poeng)

Bruk Pythons strengmetoder til å løse følgende oppgaver.

a) Les inn en tekstfil
Filen `in01.txt` inneholder norske anmeldelser.[1] Last ned filen fra emnesiden og les den inn i koden din.

b) Tell forekomster

- Bruk strengmetoden `count()` til å avgjøre hvor mange ganger bokstavsekvensen "er" forekommer i teksten. (Du skal ikke bare telle ordet "er". En forekomst av "er" i f.eks. "hjerne" eller "her" skal også telles.)
- Hvor mange ord i teksten slutter på "er"? (NB! Her er vi ute etter endelsen "er" og ikke bare en forekomst hvor som helst i et ord (dvs. at vi vil telle med "er" i f.eks. "tenker" som en forekomst, men ikke "hjerne"). I denne oppgaven trenger du ikke å håndtere tegnsetting; tell bare ordene som slutter på "er" og overse de som slutter med f.eks. "er," eller "er.". Forekomster av "er" etterfulgt av punktum, komma eller annen tegnsetting kan altså ignoreres. Hint: Du kan bruke strengmetodene `split()` og `endswith()`.

c) Hent ut deler av et ord

- Bruk slice-notasjon til å lage en liste som inneholder de to siste bokstavene fra hvert ord i teksten.
- Sett sammen listen med endelser til en streng av endelser med mellomrom mellom. Hint: Du kan se på Python's metode `join()`.

4 Enkel tokenisering (20 poeng)

Vi ønsker nå å få tilgang til alle linjene og ordene i teksten.

- Les inn filen `in01.txt` som en liste bestående av linjer ved hjelp av en for-løkke. Hvor mange linjer består teksten av? Skriv ut resultatet til terminalen.
- Utvid løkken med funksjonalitet for å telle ord. Hvor mange ord består teksten av? Vi antar bare at ordene i hver linje er adskilt av mellomrom. Du trenger altså ikke å tenke på tegnsetting her. Her skal du benytte deg av en teller som økes for hver runde i løkken. Skriv ut resultatet til terminalen.

5 Feilanalyse (30 poeng)

a) Tokenisering fra oppgave 4
Ta utgangspunkt i tokeniseringsoppgaven fra oppgave 4. Les inn filen inO1.txt og skriv ut resultatet (ett ord per linje) til en tekstfil. Du kan f.eks. kalle den nye filen `tokenisering.txt`.

b) Inspisér resultatet
Beskriv feilene du finner og gi minst tre eksempler på ulike typer feil. Blant annet, er dine tokens fullstendige ord? Blir tegnsetting og anførselstegn skilt ut fra dine tokens?

[1] Tekstene er hentet fra NoReC-korpuset: https://github.com/ltgoslo/norec