Status Quo efter lektion 1

I har alle fået installeret eller adgang til et environment hvori mapperne `lt-course` og `langtech` findes.

- `lt-course` indeholder et checkout af https://github.com/Oqaasileriffik/lt-course og kan opdateres med `cd ~/lt-course` efterfulgt af `git pull`
- `langtech` indeholder Kalaallisut-analysatoren og kan opdateres ved at køre `~/lt-course/lecture01/scripts/update-svn.sh` - dette kan tage op til 40 minutter at køre færdig hvis der er betydelige ændringer siden man sidst har opdateret - I behøver nok ikke opdatere før Trond tager over i april

I har alle kørt disse kommandoer, her gengivet med kort forklaring før:

- Vis hvilke filer/mapper der er i den mappe terminalen står i
```
ls
```
- Skift til mappen lt-course
```
cd lt-course
```
- Skift til mappen lecture01
```
cd lecture01
```
- Skift til mappen corpus
```
cd corpus
```

- Læs filen Ataqqinartuaraq.txt og vis den i terminalen
```
cat Ataqqinartuaraq.txt
```
- … og vis den i terminalen, men med pause per side
```
cat Ataqqinartuaraq.txt | more
```
- … og vis hvor mange linier, ord, og bytes den indeholder
```
cat Ataqqinartuaraq.txt | wc
```

- Læs filen Ataqqinartuaraq.txt og erstat alle spaces med linieskift, og vis med pause per side
```
cat Ataqqinartuaraq.txt | tr ' ' '\n' | more
```
- … og sorter linierne alfabetisk
```
cat Ataqqinartuaraq.txt | tr ' ' '\n' | sort | more
```
- … og vis kun den første af hver identisk linie
```
cat Ataqqinartuaraq.txt | tr ' ' '\n' | sort | uniq | more
```
- … og vis hvor mange unikke ord den indeholder
```
cat Ataqqinartuaraq.txt | tr ' ' '\n' | sort | uniq | wc
```

- Læs filen Ataqqinartuaraq.txt og opsummer forekomster af unikke ord
```
cat Ataqqinartuaraq.txt | tr ' ' '\n' | sort | uniq -c | more
```
- … og sorter efter antal forekomster, men med største tal (dvs. mest frekvente ord) først
```
cat Ataqqinartuaraq.txt | tr ' ' '\n' | sort | uniq -c | sort -n -r | more
```

Allerede her har man en brugbar grovsortering af mest frekvente ord i et corpus med antal forekomster. Men, hvis man vil skrælle punktum, komma, og andre tegn fra, så kan man tilføje dem til listen af tegn i `tr` som skal fjernes:

```
cat Ataqqinartuaraq.txt | tr '., !"”:?-' '\n' | sort | uniq -c | sort -n -r | more
```

Forklaring af hver del af kommandoen:

1. `cat Ataqqinartuaraq.txt` Læs filen Ataqqinartuaraq.txt
2. `|` *...og send den videre til...*
3. `tr '., !"”:?-' '\n'` Erstat punktum, komma, space, udråbstegn, dobbelt-quote, smart-quote, kolon, spørgsmålstegn, og bindestreg med et linieskift
4. `|` *...og send det videre til...*
5. `sort` Sorter linier alfabetisk
6. `|` *...og send det videre til...*
7. `uniq -c` Opsummer antallet af identiske linier
8. `|` *...og send det videre til...*
9. `sort -n -r` Sorter numerisk (`-n`) og med største først (`-r`)
10. `|` *...og send det videre til...*
11. `more` Vis outputtet på terminalen i sider man kan overkomme at læse

Hvorfor virker det?

Mange programmer og kommandoer i terminalen er linie-baserede. E.g. `sort` og `uniq` arbejder med hele linier ad gangen. Så for at bruge de værktøjer kan vi ikke bare tage en tekst og sortere ordene i den - vi skal først transfomere teksten fra ord til linier af ord.

Men her rammer vi endnu et problem: Hvad er et ord? Den nemme løsning er at sige et ord er omsluttet af space. Dette giver et hurtigt overblik, men ord som slutter på punktum eller komma bliver set som forskellige fra ord uden tegn. Man kan vælge at slette alle tegn, men så rammer man problemet med at ord der er delt med bindestreg vil blive skrevet sammen til ét. Så alle uønskede tegn skal laves om til spaces for at ordentligt dele ordene, og fordi vi ved vi alligevel gerne vil have hvert ord på en linie for sig selv, så kan vi erstatte alle tegn og spaces med linieskift i et hug med en `tr` som erstatter tegnene `'., !"”:?-'` med `'\n'`.

Så sender vi strømmen af ord-per-linie til `sort` som, hvis man ikke siger andet, sortere linier i alfabetisk rækkefølge. Tomme linier kommer alfabetisk først, så hvis man kun kigger på dette led vil der være mange sider med ingenting.

Herefter kommer `uniq` som, hvis man ikke siger andet, fjerner linier hvis den foregående linie er identisk. Da linierne er blevet sorteret, ved vi på dette niveau at alle identiske linier vil komme i klumper, så `uniq` kan udføre sit arbejde maksimalt. Men, at fjerne linierne giver kun et overblik af hvilke unikke ord der eksisterer i teksten. Nogen gange er det det resultat man gerne vil have, hvis man bare skal kigge på om bestemte ord overhovedet bliver brugt.

Vi vil dog gerne have en frekvensliste af de mest hyppige ord. Heldigvis har `uniq` et flag `-c` (count) der optæller antal forekomster i stedet for bare at fjerne dem.

Så nu har vi faktisk en frekventliste hvor hvert ord er optalt med hvor mange gange det forekommer i teksten. Men rækkefølgen er alfabetisk, fordi det er hvad sidste `sort` gav os. Nogen gange skal man bruge listen alfabetisk, så det kan være ok at stoppe her.

Vi vil dog gerne have listen sorteret så de mest hyppige ord kommer først. Heldigvis er outputtet af `uniq -c` linier hvor antallet af forekomster står først på linien, og `sort` har flag `-n` der sorterer numerisk i stedet for alfabetisk. Og fordi `sort` sorterer i stigende rækkefølge så de mindste tal kommer først, sætter vi også `-r` på for at ændre det til faldende rækkefølge så de største tal kommer først. Resultatet er en frekvensliste af ord hvor de mest hyppige står først.

Til utroligt meget leksikografisk arbejde er dette alt man behøver. En frekvensliste af et corpus giver rigtig meget information om teksterne i corpusset.

Opgave til onsdag d. 5

Kig på og sammenlign frekvenslister for alle 3 corpora i `lt-course/lecture01/corpus/` mappen.

Noter & Hints

`cd` er en af de mest vanskelige kommandoer. Det at navigere rundt i mapper i en tekstbaseret terminal er meget svært for mange. Det er helt normalt at blive væk. Hvis i ikke ved hvor i er, kan i altid gå tilbage til roden ved at køre `cd` alene uden argumenter.

Hvis man ved hvor man vil hen kan man sætte mapper sammen i `cd`. E.g. i stedet for at køre 3 separate `cd lt-course` og `cd lecture01` og `cd corpus` kan man køre 1 samlet `cd lt-course/lecture01/corpus`.

Mappen `~` betyder ens rod-mappe eller hjemme-mappe (home). Lige meget hvor man faktisk står, kan man altid hoppe til noget relativt til `~`. E.g., selvom man står i `/tmp` kan man køre `cd ~/lt-course/lecture01/corpus` for at hoppe til den mappe.

Tab-tasten ↹ på tastaturet (normalt over caps lock) kan bruges til at auto-complete filnavne og kommandoer. E.g. hvis man i `corpus`-mappen skriver `cat At` og trykker på ↹ så finder terminalen selv ud af at skrive `cat Ataqqinartuaraq.txt`.

Hvis der ikke sker noget når man trykker ↹, så tryk igen. Det kan være der er flere filer der opfylder hvad man har skrevet, og hvis man trykker ↹ 2 gange vises de på terminalen. E.g. hvis man i `corpus`-mappen skriver `cat A` og trykker på ↹ 2 gange så vises der `Ataqqinartuaraq.txt` og `Aviscorpus.txt`. Herefter skal man så selv skrive indtil resten er unikt før man trykker ↹ igen.

Benyt `ls` flittigt. Det skader aldrig at se hvilke filer der er i den mappe man står i. `ls` tager også mange flags, hvor den mest brugte er `-l` (long) som viser listen af filer og mapper vertikalt med mere information.

`more` afsluttes ved at trykke tasten q.

Generelt kan man afslutte alt ved at trykke tasterne CTRL og c. I mange manualer skrives CTRL+c som `^C`, fordi CTRL vises som `^` når den forekommer i terminalen. Så hvis terminalen gør noget mærkeligt eller i gerne vil tilbage til en tom prompt, tryk på CTRL+c.

Mht. flags, benyt `-h` eller `-?` eller `--help` eller `man` flittigt for at se hvad kommandoerne kan.