Failidest otsimine

Failidest märgijada otsimiseks on grep-käsk. See on võrreldav kirjakeele korpuse nn otsiauguga: siin on võimalik koostada samasuguseid päringuskeeme nagu otsimootoriski, ainult tuleb jälgida õiget sisendit ja väljundit. Siin kehtivad ka samad erisümbolid, mis kirjakeele korpuse otsiaugus.

grep
otsib välja kõik vastavat stringi (sõna, sõnaosa) sisaldavad READ (NB! Reaks loetakse seda, mis on eraldatud reavahetusega, mitte seda rida, mida ekraanilt näete)

grep 'maja' otsib välja kõik read, kus esineb sõna *maja*
grep -c 'maja' esitab ridade arvu, kus sõna *maja* esineb
grep -v 'maja' otsib välja need read, kus ei esine sõna *maja*
grep -2 'maja' jätab järjendit *maja* sisaldava rea ette ja järele veel kaks rida

Näide
Murdekorpuses on keelejuhi teksti alguses alati märgen < who=“KJ>. Kui keelejuhte on mitu, siis vastavalt KJ ja KJ2. Et otsida korpusest AINULT keelejuhi tekstis olevaid sõnu (st keelejuhi kõnevoore), maksab enne täpsustavaid otsinguid välja otsida need read, kus keelejuht räägib, seejärel sealtsamast edasi need read, kus on otsitav tekstilõik, näiteks sõna *vot*:

cat KOD* | grep 'who=.KJ' | grep ' vot ' (punkt tähendab siin seda, et seal peab olema mingi suvaline sümbol, jutumärkide kasutamisega otsingusõne sees võib tekkida probleeme)

Taolise pärimise järel on ainus probleem, et vastuseks on read, ühe rea pikkus võrdub kogu kõnevooru pikkusega. Selleks, et ridu natuke lühemaks saada, tuleks need tükeldada. Sellest veidi hiljem.

Vihje:
"Toru" puhul tuleb alati arvestada sellega, et ühe käsuga sooritatav väljund on ühtlasi sisendmaterjal järgmise käsu täitmiseks. Seepärast on käskude sooritamise järjekord väga oluline. Kui te täpselt ei tea, mida üks või teine käsk teeb (st missugune väljund sealt tuleb, mida edasi töödelda kavatsete), tasub töö kõiki etappe üksteise järel läbi katsetada. Sellega hoiate tegevust oma kontrolli all ning leiate hõlpsamalt üles vead, mis võivad tekkida. Eelmises näites võiks eri etappe ükshaaval katsetada nii:

cat KOD* | more siit näed, milline on KOD tekstid, mida edasi töötlema hakkad
cat KOD* | grep 'who=.KJ' | more näed, et grep-käsk toimib ja milline on väljund
cat KOD* | grep 'who=.KJ' | grep ' vot ' ka siia võib lõppu panna | more , siis on mugavam sirvida.

NB! more-käsu katkestamiseks Cntr+c.

Erisümbolid
töötavad grep- ja sed-käsuga

.	üks suvaline märk
.*	mistahes sümbol null kuni lõpmatu arv kordi
x*	x esineb null kuni lõpmatu arv kordi
[a-z]	inglise tähestiku väiketähed
[A-Z]	inglise tähestiku suurtähed
[0-9]	kõik numbrid
[^a]	kõik märgid, välja arvatud *a*
[^a-zA-Z]	mistahes sümbol, v.a tähestiku tähed
x{2,5}	otsib märgijadasid, kus x-i esineb minimaalselt 2, maksimaalselt 5 korda järjest
|	sooritab kaks otsingut, mis eraldatud |-ga
x{2}	otsib märgijadasid, milles x-i esineb vähemalt 2 korda järjest (maksimum pole piiratud)
x+	otsib märgijadasid, milles x esineb vähemalt 1 korra (maksimum pole piiratud)
\?	kaldkriips tühistab järgneva märgi erisümbolitähenduse
^	rea algus
$	rea lõpp

tr
asendab sümboleid
tr 'a' 'A' asendab kõik väikesed a-d suurte A-dega
tr 'abc' 'efg' asendab kõik *a*-d *e*-dega, kõik *b*-d *f*-idega ja kõik *c*-d *g*-dega
tr '[A-Z]' '[a-z]' asendab kõik suurtähed väiketähtedega
tr -d 'a' kustutab tekstist kõik *a*-d
tr -d '0-9' kustutab kõik numbrid
tr ' ' '\012' asendab kõik tühikud reavahetusega
tr -s '\012' kustutab tekstis kõik korduvad reavahetused, s.t tühjad read

Näide:
Oletame, et on vaja otsida Võru murde tekstidest välja kõik sõnad (mitte read!), mis sisaldavad q-d (larüngaalklusiili). Selleks tuleks eelnevalt 1) otsida sobivad tekstid; 2) otsida neist välja ainult keelejuhi kõnevoorud; 3) asetada iga sõna eraldi reale, et saaks otsida ainult vajalikke sõnu, mitte ridu (see tähendab, et asendan tühikud reavahetusega); 4) otsida välja q-d sisaldavad read.
Saame päringurea (otsin praegu välja kõik Võru tekstid, selleks liigun Võru murde kataloogi):
cd Vorumurre
cat *.txt | grep 'who=.KJ' | tr ' ' '\012' | grep 'q' | more
Tulemust vaadates näeme, et vastuste hulgas on ka sõnaühendeid, mille osiste vahel on kokkuhäälduse märk =. Parem oleks ka need asendada reavahetusega:
cat *.txt | grep 'who=.KJ' | tr ' ' '\012' | tr '=' '\012' | grep 'q' | more

sed
sobib kasutada siis, kui asendada on vaja märgijada, mitte üksiksümboleid
asendab ühe stringi (sümboliterea, nt sõna) teisega
sed 's/maja/auto/g' asendab kõik *maja*-sõnad sõnaga *auto* (s ja g märgivad, et tegemist on regulaarse asendusega)
sed 's/ä/ä/g' asendab kõik html-i koodis ä-d tavaliste ä-dega
sed 's/(\.\.\.)/ /g' asendab (...) tühikuga. sed-käsus on kaldkriips punktide ees selleks, et võtta punktilt ära erisümboli tähendus: sed 's/(...)/ /g' asendaks ka muud sulgude vahel olevad kolm sümbolit tühikuga. (See pole probleem, kui tekstis sulgude vahel muid kolme sümboliga järjendeid pole kasutatud.)

Näide

Murdekorpusest keelejuhi tekstist sõnade otsimisel on voorud üldjuhul tülikalt pikad. Selleks, et neid lühendada, on mitmeid võimalusi. Küllaltki loogiline koht nende lühendamiseks on pikkade pauside koha pealt: pikad pausid on märgitud (...). Kuna tegemist on märgijadaga (mitte üksiksümboliga), ei saa seda otse reavahetuse koodiga asendada. Mõistlik on enne (...) asendada ühe sümboliga, mida korpuses muidu pole kasutatud, seejärel see üksiksümbol tr-käsuga asendada reavahetuse koodiga. Näiteks võib (...) asendada X-iga (sed-käsuga), seejärel saab juba tr-käsuga sellesama X-i asendada reavahetusega.

cat *.xml | grep 'who=.KJ' | sed 's/(\.\.\.)/X/g' | tr 'X' '\012'

Lisaks ridade tükeldamisele on mõnikord vaja lahti saada ka veel märgendusest: vooru alguse ja lõpu märkidest, kommentaaridest.
Iga vooru algul on märgitud vooru alguse märk u ja kõneleja: <u who=KJ>
Iga vooru lõpus on vooru lõpu märk </u>

Kommentaaride alguses ja lõpus on <com> kommentaar </com>, seega on kommnetaari märkide vahel litereerija sõnu, mis võivad segi minna keelejuhi või küsitleja päris tekstiga.
Seega oleks vaja vabaneda 1) kõigepealt kommentaaridest; 2) seejärel veel järele jäänud märgenditest.

1) Kommentaaridest vabanemiseks asendame mittemillegagi (või tühikuga, maitse asi) sed-käsu abil lõigu, mis algab <com>, selle vahel on ükskõik mis, v.a < (et ei ületataks kommentaaride lõpumärki) ükskõik kui palju, seejärel on </com>. Kuna märgendis </com> / on ise sed-käsu sees väljade eristaja, on vaja tema ette panna tagurpidi kaldkriips, et talt ära võtta erisümboli tähendust:
sed 's/<com>[^<]*<\/com>//g'

2) muudest märgenditest vabanemine on juba lihtne, sest märgendite vahelt ei ole vaja kustutada. Asendame mittemillegagi (või tühikuga) järjendi < , seejärel ükskõik mis, mis poleks märgendi lõpusümbol (ja seda ükskõik kui palju), seejärel märgendi lõpusümbol >
sed 's/<[^>]*>//g'

Paneme nüüd kokku käsujärjendi, milles kustutatakse kommentaarid ning hakitakse tekst lühemaks. Suuname ta praegu eraldi faili: murded.txt
Kogu käsk: cat *.xml | grep 'who=.KJ' | sed 's/<com>[^<]*<\/com>//g' | sed 's/<[^>]*>//g' | sed 's/(\.\.\.)/X/g' | tr 'X' '\012' > murded.txt

Faili suunamise asemel võib ka sirvimiseks standardväljundisse suunata (lehitsemisega):
cat *.xml | grep 'who=.KJ' | sed 's/<com>[^<]*<\/com>//g' | sed 's/<[^>]*>//g' | sed 's/(\.\.\.)/X/g' | tr 'X' '\012' | more

Nüüd on teksti niipalju puhastatud, et võib juba midagi asjalikku otsima hakata ☺