Natuurlijke taalmodellen en interfaces

Eszter Fodor & Sharon Gieske

April 8, 2012

Question 1

grep '^A:' Deze command zoekt naar een regel die met de empty string (^) begint gevolgd door de letter *A* en dubbelepunt.

sed 's/^A: //': Deze command verwijdert de empty string gevolgd door de letter A en een dubbelepunt *(^A:)* van de door grep gevonden regels. De command *¿ ovis-trainset.txt* definieert de output file.

Question 2

*grep -v '^$':* Deze command zoekt naar niet lege regels. *-v* betekent dat er juist *niet* naar de gespecificeerde regels moet worden gezocht, in dit geval een lege regel (^$).
sed 's/\s \+ / \n/g': Verwijdert aan het eind van alle regels de spatie (\s) en maakt er een newline van.
uniq -c: Haalt duplicates uit de input en telt (*-c*) hoeveel keer iedere regel voorkomt.
sort -g -r -k 1: Sorteert de resultaten aan de hand van de hoeveelheid voorkomens en print deze in reversed volgorde (van het hoogste getal naar het laagste). *-g*: sorteer aan de hand van getallen, *-r*: reverse de volgorde van sorteren, *-k 1*: sorteer op de eerste key.

Question3

Zipf distribution. ??

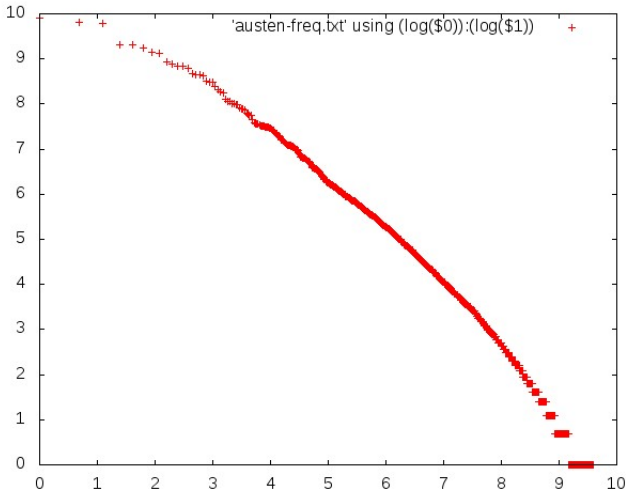
Question 4

Met de volgende opeenvolgende commands kan de word-frequency van austen.txt geplot worden:

```
less austen.txt | sed 's/\s\+/\n/g' | grep -v '^$'
        | sort | uniq -c | sort -g -r -k 1
        > austen-freq.txt

gnuplot
gnuplot> plot 'austen-freq.txt'
        using (log($0)):(log($1))
        gnuplot>set terminal jpeg
gnuplot> set output 'austen.jpg'
gnuplot> plot 'austen-freq.txt' using
        (log($0)):(log($1))
```

Als output plaatje krijgen we dan:



De twee plaatjes van de word-frequencies lijken vrijwel identiek.