Gustav Leunbach

Om automatisk orddeling.
Forslag til en undersøgelse.

I flere systemer til automatisering af trykning og tekstfremstilling i det hele taget indgår automatisk orddeling for at undgå for store forskelle i linjelængderne, og især i diskussionen om læseligheden af avistryk spiller de af og til bizarre virkninger af denne orddeling en vis rolle; aviserne hævder - med rette - at det kun drejer sig om få procent af samtlige orddelinger, men det er ting som falder i øjnene.

De anvendte algoritmer er vistnok forretningshemmeligheder, og en opgave kunne være at forsøge at dechiffrere dem udfra den producerede sats. Men det forslag jeg nu fremsætter går i en anden retning, nemlig at sammenligne forskellige algoritmers virkning på den samme simulerede eller naturlige tekst.

Kvaliteten af en orddelingsalgoritmes produktion er sammensat af to komponenter: akceptabiliteten af de foretagne delinger og variationen i linjelængde. Der er et komplementært forhold mellem disse to; hvis man slet ikke deler må man akceptere den maximale variation i linjelængder; omvendt kan man slutte enhver linje ved position nr. 60 hvis man til gengæld finder sig i at ord deles hvor som helst.

Akceptabiliteten af foretagne delinger kan ikke bedømmes efter faste retningslinjer, men må i nogen grad bero på skøn, f.ex. er retskrivningsordbogens angivelse af tilladte delinger ikke fuldt algoritmiserbar. En graduering af akceptabiliteten er meget vel tænkelig, men jeg foreslår at et ulige antal bedømmere blot får til opgave at svare ja eller nej til hver foretagen deling og at flertallet afgør. Hvis graden af overensstemmelse er for lille kan man overveje at sende alle bedømmerne til omskoling.

Variabiliteten af linjelængder er et numerisk kriterium, men ikke endimensionalt, idet en algoritme både producerer mange linjer med lidt under standardlængden og enkelte som afviger stærkt; en rimelig sammenvejning kunne være det gennemsnitlige "tab" i forhold til standardlængden kombineret med et absolut forbud mod at overskride denne.

Udover algoritmers produktion kan de også bedømmes på deres kompleksitetsgrad, deres "længde". Der kan sikkert skabes enighed om at en algoritme som består i en opregning af alle tilladte delinger af alle ubøjede og bøjede ord i et korpus er for lang; omvendt kan man få en ultrakort og ubrugelig algoritme ved at sige at ethvert ord kan deles efter hvert tredje bogstav. Den første indeholder for mange konstanter,

Om automatisk orddeling: Forslag til en undersøgelse
Gustav Leunbach
Proceedings of NODALIDA 1979, pages 43-45

den anden for få; en opregning af hvilke bogstavsymboler der angiver vokaler vil de fleste nok betragte som et tilladeligt sæt konstanter. Hvis man ønsker at præcisere "længden" af algoritmer skal man vel tænke på antallet af linjer i en kodning i et vist assemblersprog.

Det giver et stort spild at måle en algoritmes produktion på de ord som tilfældigvis kommer til at afslutte linjerne i en normal sætning af en tekst, og mit forslag indebærer en simulation på grundlag af en sandsynlighedsbetragtning for hvilke ord som bliver udsat for at skulle deles.

Grundmaterialet er en liste over ordformer med deres frekvenser i et korpus sorteret efter ordlængde, således at 1-bogstavs ord (f.ex. ordformerne i, ø og å) tilsammen har en frekvens på a_1, 2-bogstavs ord tilsammen a_2 osv. Lad os antage at der længste ordform har 27 bogstaver. Sandsynligheden for at det næste ord har længden j kan da til enhver tid sættes til

$$b_j = a_j/(a_1+a_2+\ldots+a_{27}).$$

(Virkningen af at ord ikke kommer i tilfældig rækkefølge er svær at beregne, men er antagelig lille for de variable det her drejer sig om.)

Hvis der på et givet tidspunkt er k positioner tilbage på linjen (medregnet det mellemrum som skal følge efter sidste ord) vil der da være sandsynligheden

$$c_k=b_k+b_{k+1}+\ldots+b_{27}$$

for at det næste ord ikke kan stå der og altså må søges delt. Det kan da beregnes at sandsynligheden for at delingsbehovet opstår netop når der er k positioner tilbage er omtrent proportional med c_k, og sandsynligheden for at det ord der skal deles er af længden j bliver en konstant gange $c_k \cdot b_j$ (for hvert $j \geq k$).

Simulationen går herefter ud på at der ved tilfældige tal trækkes er tal k med sandsynligheden c_k gange normeringskonstanten og derefter et ord blandt alle der mindst fylder k bogstaver med sandsynligheder bestemt af ordenes frekvens.

Dette ord søges så delt sådan at højst k-2 bogstaver står før delestregen (denne skal også kunne være på linjen); tabet bliver k-2-i hvis i bogstaver står før delestregen og k hvis deling ikke lader sig gøre. Resultatet af en algoritme udtrykkes dels ved fordelingen af tabene på værdierne mellem 0 og 27, dels ved procenten af uakceptable delinger efter bedømmernes flertalsafgørelse. Forskellige algoritmer

kan bedømmes på det samme sæt af ord-antal-par; de foretagne delinger kan præsenteres for bedømmerne i randomiseret orden, og de tilfælde hvor alle algoritmer har givet samme deling behøver slet ikke bedømmes, medmindre man foruden sammenligningen også ønsker at bestemme algoritmernes akceptabilitetsniveau.

Jeg har selv en algoritme parat som jeg tilbyder til sammenligning med andres. Det korpus jeg disponerer over er måske for lille, men andre kan vel levere et bedre.
