



UPPSALA
UNIVERSITET

Grundläggande Textanalys VT 2014

Språkgranskning (1)

Eva Pettersson

eva.pettersson@lingfil.uu.se





UPPSALA
UNIVERSITET

Översikt

- Denna gång
 - Stavningskontroll
 - Allmänt om stavningskontroll
 - Feligenkänning
 - Felkorrigering
 - Samarbetsuppgift
- Nästa gång
 - Grammatikkontroll
 - Stilkontroll
 - Kontrollerat språk
 - Språkgranskningssystem, med fokus på MS Word och Granska



Vad förväntas av det ideala stavningskontrollprogrammet?

- Känna igen och larma för alla felstavade ord
- Känna igen och acceptera alla rättstavade ord
- Ge ett korrekt rättningsförslag för alla felstavade ord

Realistiska förväntningar på stavningskontrollprogrammet

- Känna igen och larma för ~~alla~~ *de mest frekventa och/ eller lättidentifierade* felstavningarna
- Känna igen och acceptera alla rättstavade ord, *som är tillräckligt frekventa i språket*
- Ge ett ~~korrekt~~ *sannolikt* rättningsförslag för alla felstavade ord



UPPSALA
UNIVERSITET

Stavningskontrollens två delar

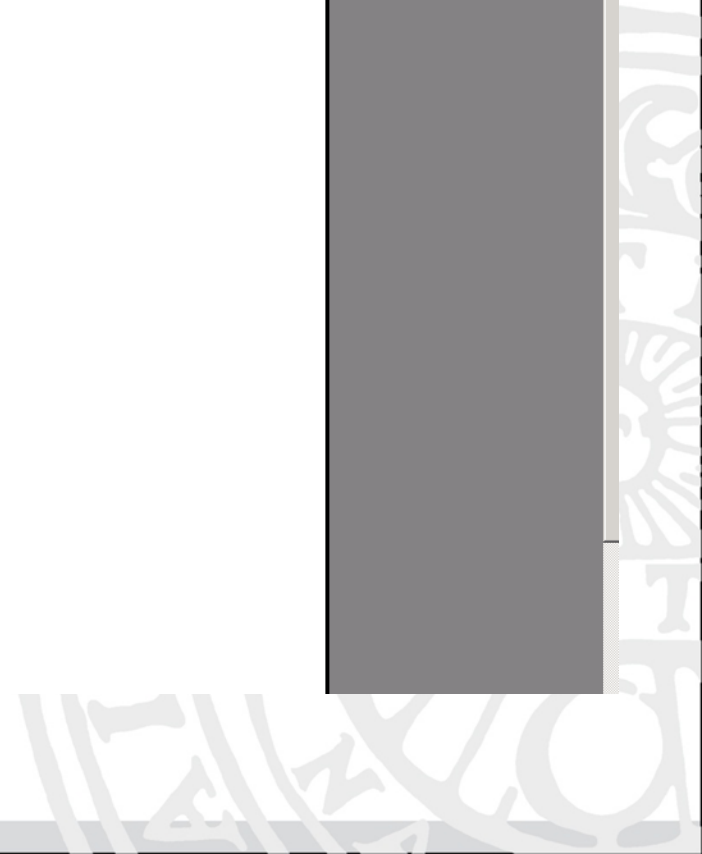
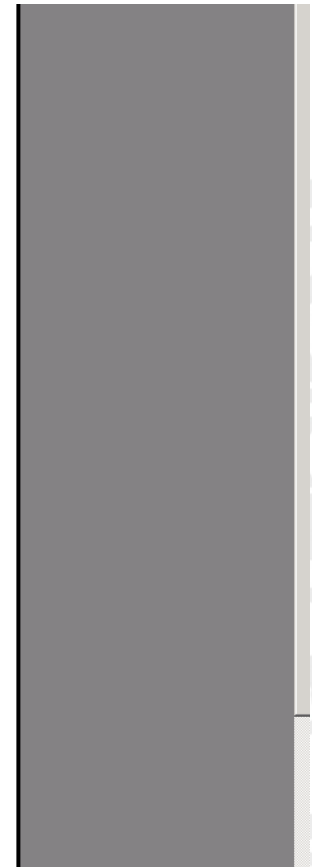
1. Feligenkänning (*error detection*)
att hitta felen
2. Felkorrigering (*error correction*)
att ge ersättningsförslag





Feligenkänning i Microsoft Word

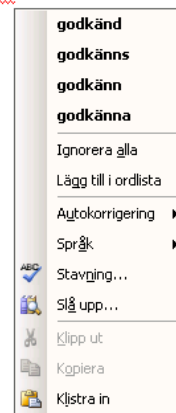
Den här stavningen är inte godkänd.





Felkorrigering i Microsoft Word

Den här stavningen är inte godkänd.





UPPSALA
UNIVERSITET

FELIGENKÄNNING





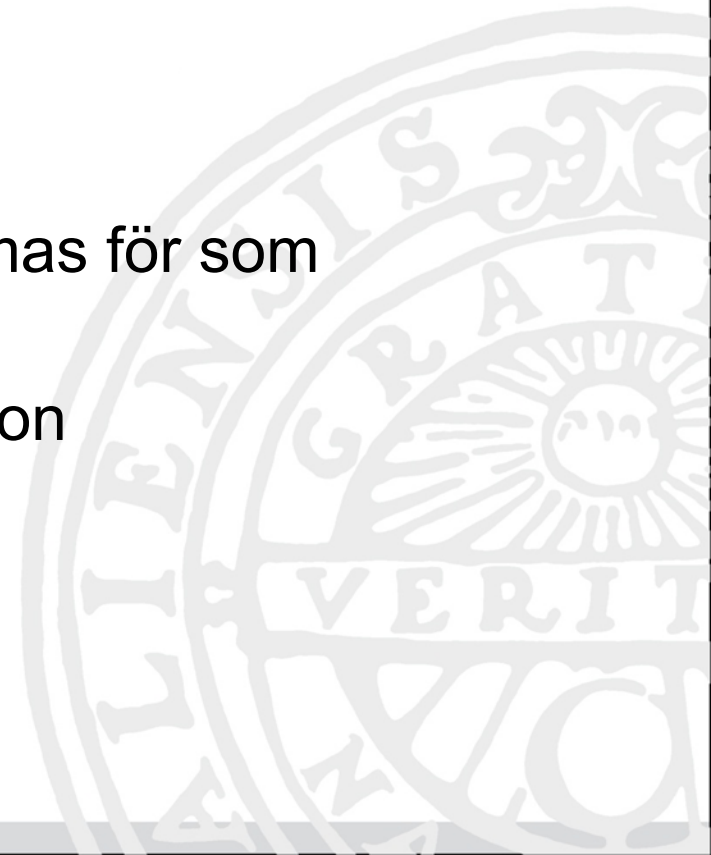
Feligenkänning

- Isolerade ord
 - Skrivfel som resulterar i icke-ord: *och* → *coh*
- Ord i kontext
 - Stavfel som resulterar i riktiga ord (real word errors)
*jag er dålig på att stava
språk teknologi är kul*
 - Bättre korrigeringsförslag
*det är **sårt** att stava*
Word föreslår: såret
 svårt
 sårat
 såt
 sått



Feligenkänningsstrategier

- Trigram av tecken
 - Larmar för ovanliga teckenkombinationer
 - Används främst inom OCR
- Lexikon
 - Ord som saknas i lexikonet larmas för som felstavningar
 - Fullformslexikon eller stamlexikon





Svagheter med lexikonmetoden

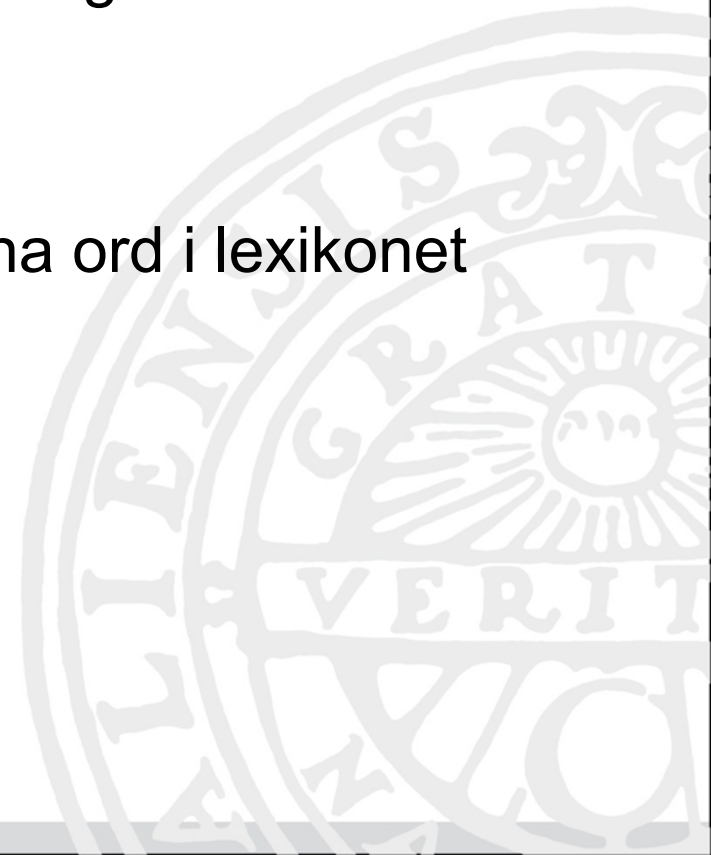
- För stort lexikon ger låg täckning
 - många fel missas (t.ex. *verv*, *boke*)
- För litet lexikon ger låg precision
 - många falska alarm
 - kan lura skribenten att till exempel särskriva
- Omöjligt att lista alla ord i lexikonet – språket är produktivt





Komplement till lexikonmetoden

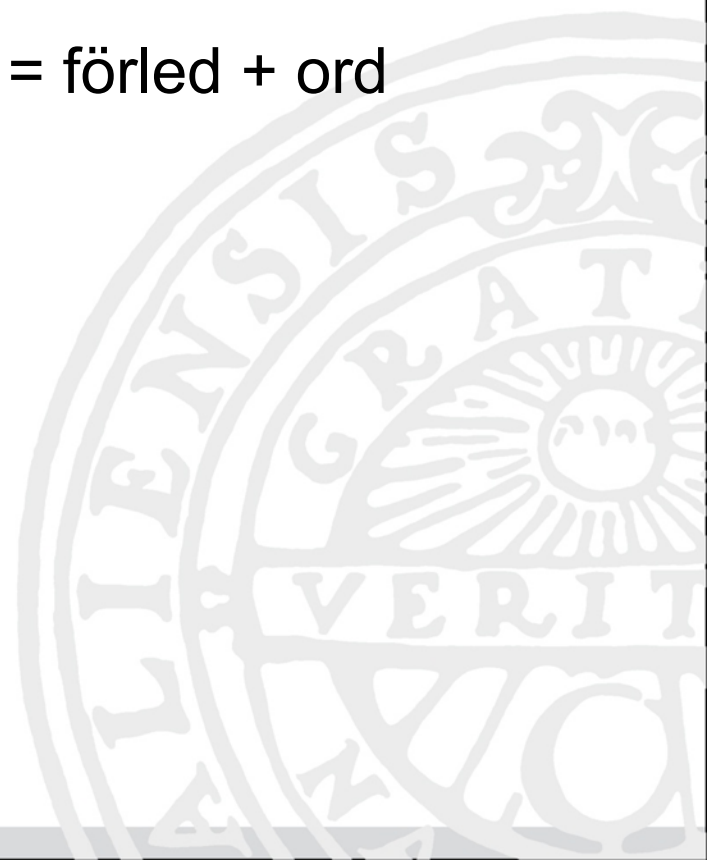
- Morfologiska regler för avledningar
- Regler för att hantera sammansättningar
- Egennamnsigenkänning
- Domänspecifika lexikon
- Tillåta användaren att lägga till egna ord i lexikonet





Feligenkänning av sammansättningar

- Basstrategi: sammansättning = ord + ord
 - *dator + lingvistik = datorlingvistik*
- Förfinad strategi: sammansättning = förled + ord
 - *flicka + klänning = flickklänning*
 - *äpple + paj = äppelpaj*
 - *kvinn + parti = kvinnoparti*
 - *cigarr + rök = cigarrök*



För- och nackdelar med sammansättningsanalys

- Minskar antalet falska alarm (bättre precision)
- Kan öka antalet missade fel (sämre täckning)
Missade fel i Word97 (åtgärdat i senare versioner):

kotakt

makelera

medalg

cykelsäll

särkskilt

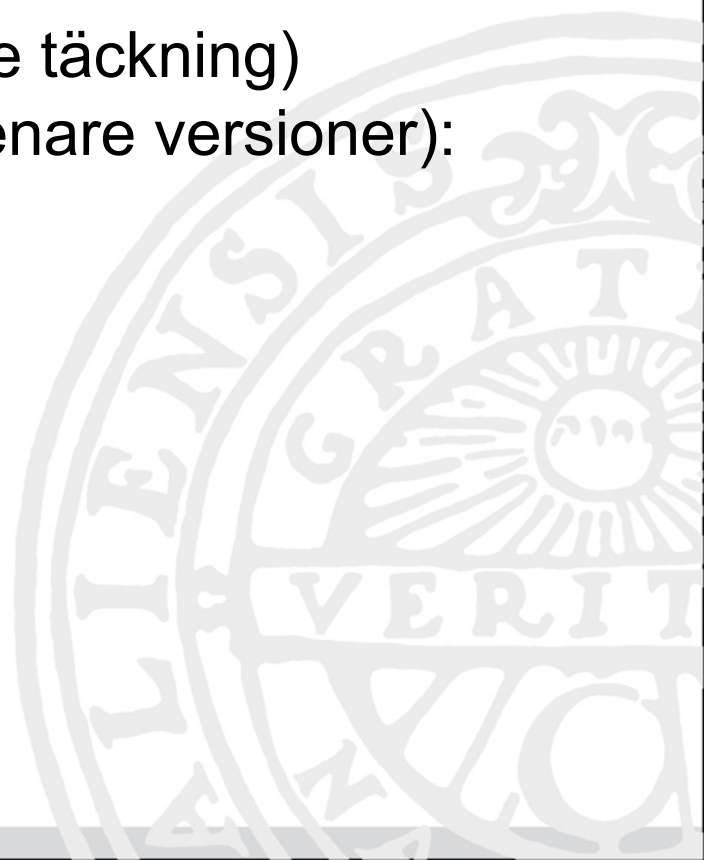
kontakt

makulera

medalj

cykelställ

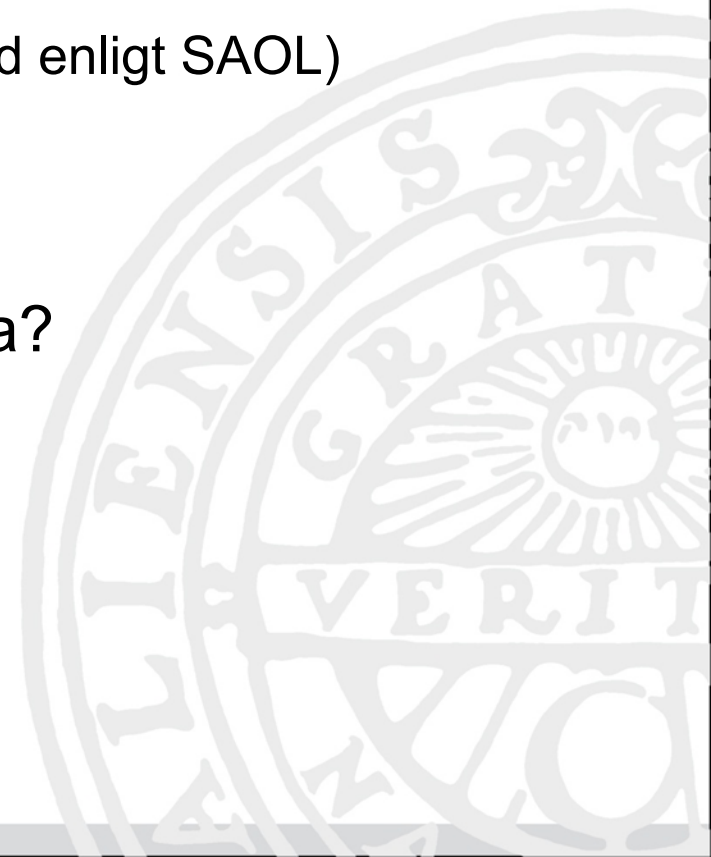
särskilt





Att fundera kring

- Hur göra med sällsynta/fackspråkliga ord? Kan ligga nära felskrivningar av frekventa ord...
 - *verv/värv* (verv = kraft, livfullhet, glöd enligt SAOL)
 - *boke/boken* (boke = bokvirke)
- Dialektala ord?
- Talspråk? Hur sträng bör man vara?
 - mej
 - direktörn
 - idag





UPPSALA
UNIVERSITET

FELKORRIGERING





UPPSALA
UNIVERSITET

Felkorrigeringens två delar

- Ta fram ett antal korrigeringskandidater
- Rangordna korrigeringskandidaterna





Feltyper

- Kompetensfel (spelling confusion)
 - Fonetiska fel: *restaurang* → *resturang*
 - Homofonfel: *gott* → *gått*
- Performansfel (typographical errors/typos)
 - Insättning *språkteknologii*
 - Borttagning *spåkteknologi*
 - Substitution *språkteInologi*
 - Transposition *spårkteknologi*



Feltyper (forts)

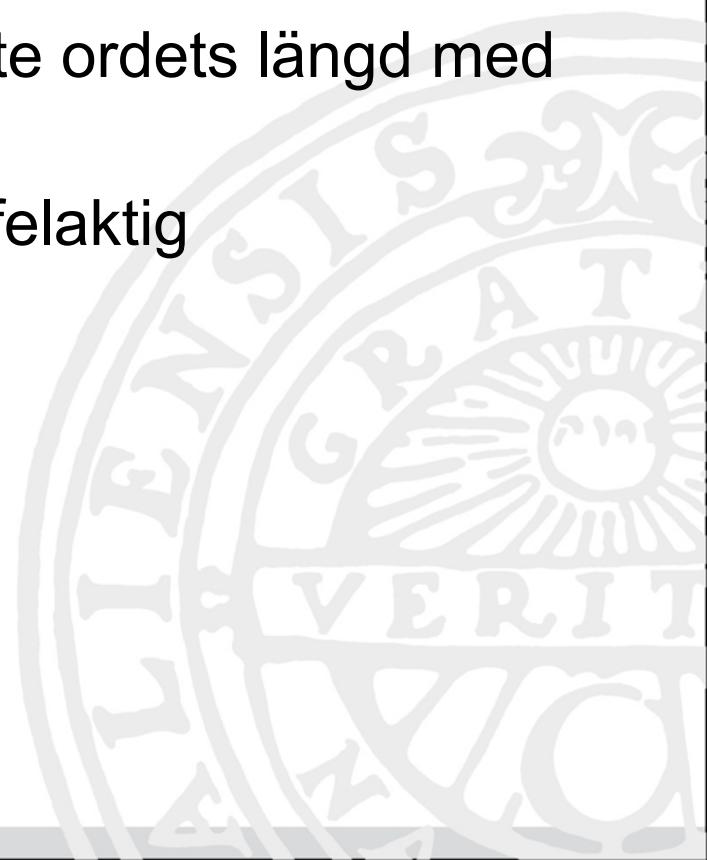
- Kompetensfel eller performansfel?
tunnt
- Oftast inte nödvändigt att veta om kompetensfel eller performansfel
- Kan ge bättre korrigeringsförslag om man tar hänsyn till feltyp

hemta kompetensfel: *hämta/hämtade*
performansfel: *hemtam*



Empiriskt grundade iakttagelser

- De flesta felstavningar är performansfel (insättning, borttagning, substitution, transposition)
- De flesta felstavningar påverkar inte ordets längd med mer än en bokstav
- Första bokstaven i ordet är sällan felaktig
- Tangenternas placering påverkar
- Bokstävernans frekvenser påverkar





Korrigeringsstrategier

- Editeringsavstånd (Minimum Edit Distance/Levenshtein Distance)
- Likhetsnycklar
- N-gramsbaserade tekniker
- Regelbaserade tekniker
- Probabilistiska tekniker





Editeringsavstånd

- Stränglikhet
- Minsta antalet editeringsoperationer som behövs för att omvandla en sträng till en annan
- Editeringsoperationer:
 - insättning
 - borttagning
 - substitution (alt. borttagning + insättning)
 - transposition (alt. substitution + substitution)



Editeringsavstånd formel

$$\text{dist}(0,0) = 0$$

$$\text{dist}(i,0) = i$$

$$\text{dist}(j,0) = j$$

$$\text{dist}(i,j) = \min$$

$$\text{dist}(i-1,j) + 1$$

$$\text{dist}(i,j-1) + 1$$

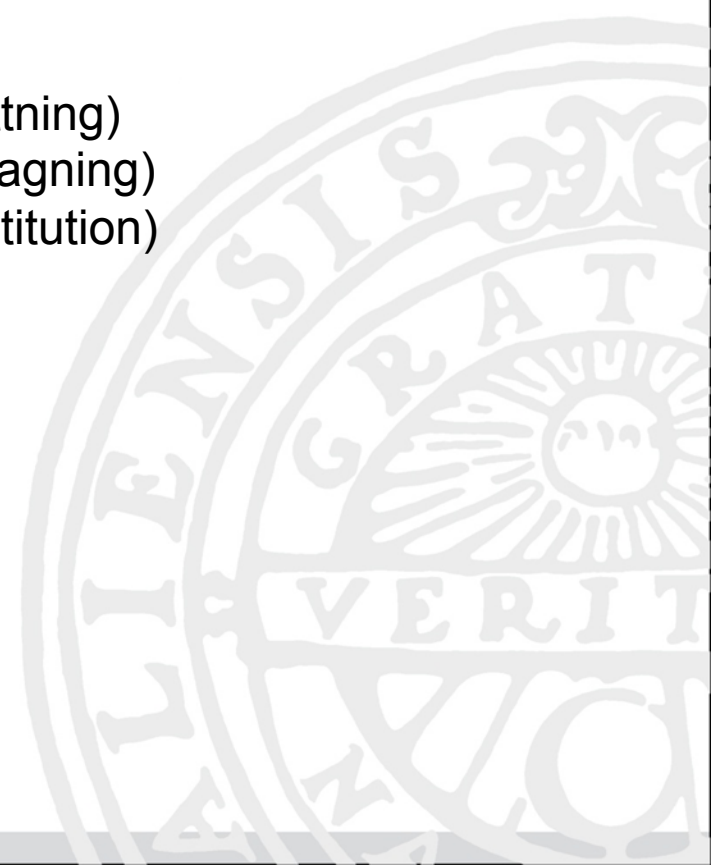
$$\text{dist}(i-j,j-1) + d(s_i, t_j)$$

(insättning)

(borttagning)

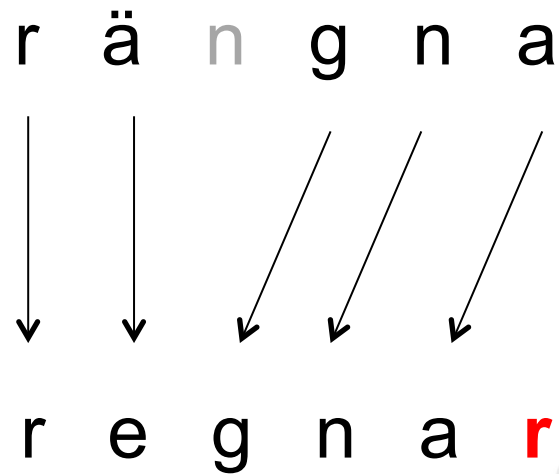
(substitution)

där i är strängen s fram till i :te tecknet, och
 j är strängen t fram till j :te tecknet, och
 $d(s_i, t_j)$ är 0 om $s_i = t_j$ och 1 i övriga fall





Editeringsavstånd illustrerat

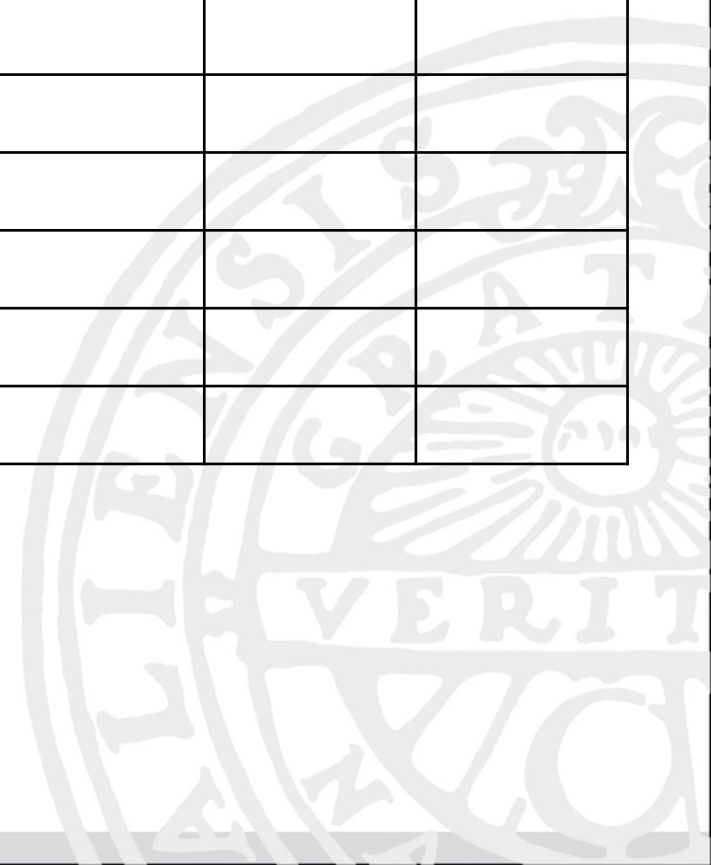


Editeringsavstånd: 3
substitution + borttagning + insättning



Editeringsavstånd: dynamisk programmering

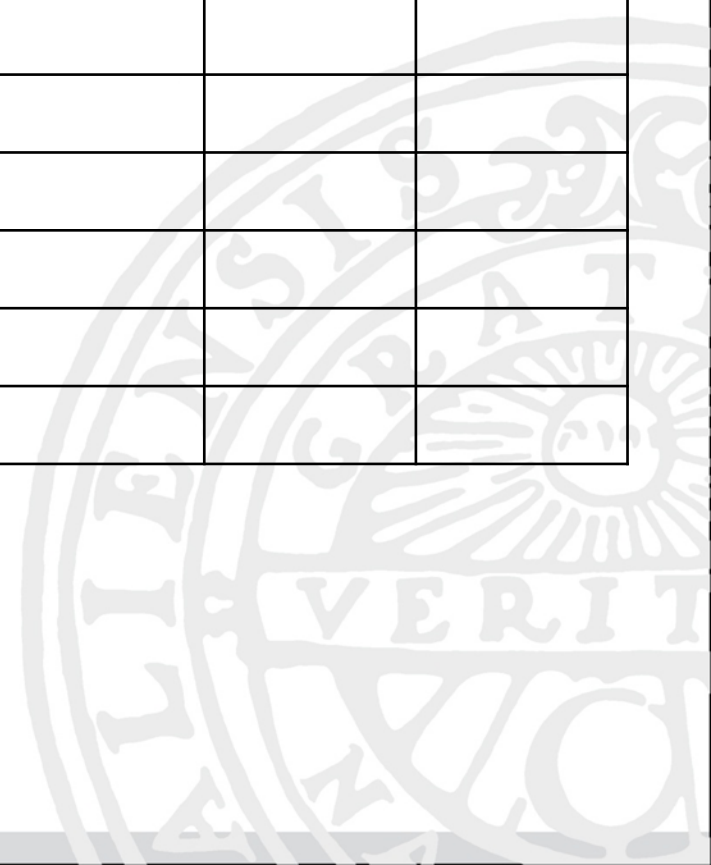
		r	e	g	n	a	r
	0						
r							
ä							
n							
g							
n							
a							





Editeringsavstånd: dynamisk programmering

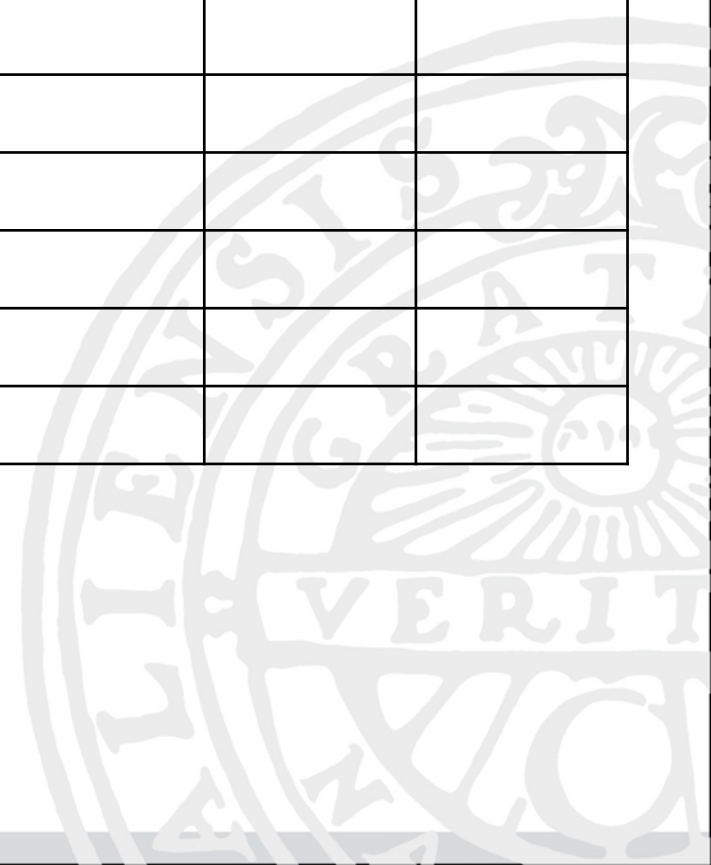
		r	e	g	n	a	r
	0	1	2	3	4	5	6
r							
ä							
n							
g							
n							
a							





Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1						
ä	2						
n	3						
g	4						
n	5						
a	6						



Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1	0					
ä	2						
n	3						
g	4						
n	5						
a	6						

kostnaden för att komma hit från min övre vänstra granne substitution	kostnaden för att komma hit från min övre granne borttagning
kostnaden för att komma hit från min vänstra granne insättning	minimum av de tre möjliga “dragen”, dvs det billigaste sättet att komma hit

Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1	0	1				
ä	2	1	1				
n	3						
g	4						
n	5						
a	6						

kostnaden för att komma hit från min övre vänstra granne **substitution**

kostnaden för att komma hit från min övre granne **borttagning**

kostnaden för att komma hit från min vänstra granne **insättning**

minimum av de tre möjliga “dragen”, dvs det billigaste sättet att komma hit

Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1	0	1	2			
ä	2	1	1	2			
n	3	2	2	2			
g	4						
n	5						
a	6						

kostnaden för att komma hit från min övre vänstra granne **substitution**

kostnaden för att komma hit från min övre granne **borttagning**

kostnaden för att komma hit från min vänstra granne **insättning**

minimum av de tre möjliga "dragen", dvs det billigaste sättet att komma hit

Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1	0	1	2	3		
ä	2	1	1	2	3		
n	3	2	2	2	2		
g	4	3	3	2	3		
n	5						
a	6						

kostnaden för att komma hit från min övre vänstra granne substitution	kostnaden för att komma hit från min övre granne borttagning
kostnaden för att komma hit från min vänstra granne insättning	minimum av de tre möjliga "dragen", dvs det billigaste sättet att komma hit

Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1	0	1	2	3	4	
ä	2	1	1	2	3	4	
n	3	2	2	2	2	3	
g	4	3	3	2	3	3	
n	5	4	4	3	2	3	
a	6						

kostnaden för att komma hit från min övre vänstra granne **substitution**

kostnaden för att komma hit från min övre granne **borttagning**

kostnaden för att komma hit från min vänstra granne **insättning**

minimum av de tre möjliga "dragen", dvs det billigaste sättet att komma hit

Editeringsavstånd: dynamisk programmering

		r	e	g	n	a	r
	0	1	2	3	4	5	6
r	1	0	1	2	3	4	5
ä	2	1	1	2	3	4	5
n	3	2	2	2	2	3	4
g	4	3	3	2	3	3	4
n	5	4	4	3	2	3	4
a	6	5	5	4	3	2	3

kostnaden för att komma hit från min övre vänstra granne **substitution**

kostnaden för att komma hit från min övre granne **borttagning**

kostnaden för att komma hit från min vänstra granne **insättning**

minimum av de tre möjliga "dragen", dvs det billigaste sättet att komma hit



UPPSALA
UNIVERSITET

Likhetsnycklar

- Ord matchas mot nycklar istället för ord
- Ord som stavas på liknande sätt har likadana (eller nästan likadana nycklar)





Likhetsnycklar: Soundex

- SOUNDEX = Indexing on Sound
- Odell & Russel, 1918 (!)
- Fonetisk likhet
 - vokaler ignoreras
 - konsonanter grupperas tillsammans om de liknar varandra fonetiskt
- Användning: Flygbokningssystem (Davidson 1962)



Soundex – algoritm

1. Behåll det första tecknet
2. Ersätt efterföljande tecken enligt nedan:

<i>a, e, i, o, u, y, h, w</i>	0
<i>b, f, p, v</i>	1
<i>c, g, j, k, q, s, x, z</i>	2
<i>d, t</i>	3
<i>l</i>	4
<i>m, n</i>	5
<i>r</i>	6
3. Ta bort alla nollor
4. Ta bort alla på varandra följande dubbletter
5. Spara de tre första siffrorna



Soundex – exempel

disapont → D215

disappoint → D215

Ersättningsförslag för **disapont**:

*disband, disbands, disbanded, disbanding, disbandment, disbandments, dispense, dispenses, dispensed, dispensing, dispenser, dispensers, dispensary, dispensaries, dispensable, dispensation, dispensations, deceiving, deceivingly, despondent, despondency, despondently, disobeying, **disappoint**, disappoints, disappointed, disappointing, disappointedly, disappointingly, disappointment, disappointments, disavowing*

N-gramsbaserade tekniker

- Stränglikhet, dvs andelen gemensamma n-gram (vanligen trigram)

$$\text{likhet}(i,j) = 2C/(n+n')$$

där n är antalet trigram i i

och

n' är antalet trigram i j

och

C är antalet trigram gemensamma för i och j



N-gramsbaserade tekniker (forts)

Hur lika är *concider* och *consider*?

##c #co con onc nci cid ide der er# r##

##c #co con ons nsi sid ide der er# r##

C (antalet gemensamma trigram) = 7

n (antalet trigram i *concider*) = 10

n' (antalet trigram i *consider*) = 10

likhet(*concider*,*consider*) = $2C/(n+n')$ = $14/20$ = **0,70**

N-gramsbaserade tekniker (forts)

Hur lika är *concider* och *cider*?

##c #co con onc nci cid ide der er# r##

##c #ci **cid ide der er# r##**

C (antalet gemensamma trigram) = 6

n (antalet trigram i *concider*) = 10

n' (antalet trigram i *consider*) = 7

likhet(*concider*,*cider*) = $2C/(n+n')$ = $12/17 \approx$ **0,71**

N-gramsbaserade tekniker (forts)

Modifierat likhetsmått:

$$\text{likhet}(i,j) = 2C/(n+n') \rightarrow \text{likhet}(i,j) = C/\max(n,n')$$

där n är antalet trigram i i

och

n' är antalet trigram i j

och

C är antalet trigram gemensamma för i och j

$$\text{likhet}(\text{con}cider, \text{con}sider) = 7/10 = \mathbf{0,70}$$

$$\text{likhet}(\text{con}cider, \text{cider}) = 6/10 = \mathbf{0,60}$$



Samarbetsuppgift

Antag att en skribent av misstag har skrivit in *käran*. Antag vidare att ett stavningskontrollprogram har kommit fram till att det rör sig om en felstavning och att möjliga ersättningsförslag är *tjäran* och *kärran*.

Hur skulle dessa ersättningsförslag rangordnas enligt det modifierade n-gramsbaserade måttet?

Hur skulle ersättningsförslagen rangordnas i termer av editeringsavstånd?

Kommentera resultatet: Är rangordningen lika bra för alla typer av skribenter?

Skulle båda alternativen finnas med bland ersättningsförslagen om man istället hade använt sig av likhetsnycklar på samma sätt som i SOUNDEX? Varför/ varför inte?



Referenser

- Markus Dickinson, Chris Brew & Detmar Meurers, 2013 (kapitel 2), *Language and Computers*
- Daniel Jurafsky & James H. Martin, 2000 (avsnitt 5.1-5.6), *Speech and Language Processing*
- Karen Kukich, 1992, *Techniques for Automatically Correcting Words in Text*
- Roger Mitton, 1996, *Spellchecking by Computer*
- Stina Nylander, 2000, *Statistics and Phonotactical Rules in Finding OCR Errors*