

Datorstöd för skrivande och grammatikkontroll

Ola Knutsson
DH2418 Språkteknologi

Innehåll

- Datorstöd för skrivande
- Olika metoder och system för grammatikgranskning.
- Granska
- Att skriva regler i Granska.

Datorstöd för skrivande

- Planering
- Textproduktion
- Revidering/granskning

Felfrekvenser

Fel i verbkedjan	21	
%Särskrivna sammansättningar	18	
%Inkongruens i nominalfraser		17
%		
Inkongruens i predikativ	4 %	
Stavfel med gram-sem konsekvens	13 %	
<i>Jag har hjort läxan</i>		
Ord saknas	13 %	
<i>Han kör ? grön bil</i>		

Vilka metoder finns?

- Regler
- Statistik och maskininlärning
- Hybrider
- Metoden kan bedömas på feldetektionsnivå eller på systemnivå.

System för andra språk än svenska

- Critique (engelska), försvann in i Word
- Storskaliga fria resurser verkar i stort sett saknas, förutom för stavning.
- Det finns kommersiella grammatikgranskare för de flesta ”stora” språken.

Parsning i Critique

Tre typer av regler:

1. 300 PLNLP-regler som försöker definiera precis de centrala grammatiska strukturer som engelska talare är överens om.
2. Regler som tar hand om flertydig utdata, det finns en metrik för hur analyser skall rangordnas.
3. Regler som hanterar när parsningen misslyckas, regler som försöker göra ett slags "rimlig" analys, s.k. Parse fitting.

Parse fitting

Parse fitting när man inte kan få ihop ett S.

Leta upp en huvudkonstituent

- a) VP med tempus och subjekt
- b) VP utan tempus, men med subjekt
- c) Fras utan verb, t.ex. NP, PP
- d) Icke-finita VP
- e) Andra

Om man hittar flera i samma kategori, väljs den längsta.

Proceduren är klar om en huvudkonstituent täcker hela meningen.

Om inte så läggs övriga konstituenten till på varje sida om huvudkonstituenten.

Fler regler -- fler analyser

Antal analyser	juni 1981	Dec. 1981
0	57 %	36 %
1	31 %	41 %
2	6 %	11 %
>2	6 %	12 %

Testmängd: 2300 meningar, upp till 63 ord, i genomsnitt 19 ord/mening

När grammatiken byggs ut får vi fler analyser och större flertydighet ...

Fenomenbaserad granskning

- Lingsofts Grammatifix i Word
- Granska
- Positiv grammatikkontroll a' la Göteborg
- Scarrie

Grammatifix (Birn 2000)

Regler i Constraint Grammar-formalism

1. Preprocessning – tokenisering, fler-ordsuttryck
2. Lexial analys - Swetwo
3. Disambiguering – tolerant version av SWECC
4. Tilldelning av @ERR och @OK
5. Fel-detektionsregler (659 stycken)
6. Om @OK finns kvar välj @OK

Exempel på en felregel i Grammatifix

Ett DT NEU @ERR @OK
högtrycksrygg N UTR @ERR @OK
passerar V PRS @ERR @OK

("<ett>" =s!@ERR)
(1C N-UTR))

Utvärdering av Grammatifix

1000504 ord från GP: 374 bra alarm och 160 falska alarm.

87713 ord från GP: Hittar 35 % (85 % eftersökta feltyper)

Finite-state parsning för grammatikkontroll

Cooper & Sofkova 1998; Sofkova 2003

% Den grova regeln (Broad grammar)

```
define      NP [dt jj* nn];
```

% de finare reglerna

```
define      Npdef[dt_def jj_def* nn_def];
```

```
define      Npind[dt_ind jj_ind* nn_ind];
```

% Den felaktiga frasen skall finnas i NP men inte i Npdef eller Npind:

```
NP – [Npdef | Npind];
```

Scarrie

- Bygger på analys från Uppsala Chart Parser
- Försöker parsa, söker därefter igenom charten med lokala regler för att upptäcka fel.
- Felkatalogen är ganska lik Grammatifix och Granska.
- Omfattande arbete med feltypologi.
- Intressant metod för att samla fel: jämförelse av ogranskad tidningstext och korrekturläst tidningstext.

15

ProbGranska: statistisk språkgranskning

Vi vill hitta oförutsägbara fel, t.ex. *Frö honom var det inget problem.*

Vi använder en hybridmetod:

- Statistik över taggtrigram (DT JJ NN 23000) från SUC (1 miljon ordklasstaggade ord)
- Lingvistisk kunskap för frastransformationer/reduktioner

En första ansats

Algoritm:

För varje position i i indataströmmen om frekvensen av $(t_{i-1} t_i t_{i+1})$ är låg i referenskorpus
rapportera fel till användaren
rapportera inget fel

Exempel på glesa data

”Det är varje chefs uppgift att ...”

Det är varje taggas som `pn.neu.sin.def.sub/obj`,
`vb.prs.akt`, `dt.utr/neu.sin.ind` och har frekvensen
noll. `dt.utr/neu.sin.ind` är ovanlig, 709
förekomster i SUC.

Ersätt en tagg med en annan liknande tagg

Vi försöker ersätta:

Det är varje chefs uppgift..

med

Det är en chefs uppgift..

(pn.neu.sin.def.sub/obj, vb.prs.akt,
dt.utr.sin.ind)

Vi får upp taggfrekvensen till 231

Frastransformationer

Exempel:

Alla hundar som är bruna är lyckliga

[NP _____]

Hundarna är lyckliga

[NP _____]

Slutsatser

- Metoden är bra på att identifiera kontextkänsliga stavfel.
- Med lingvistisk kunskap kan metoden få högre precision
- Metoden bör kunna skalas upp till n-gram över fraser och därmed bör mer strukturella fel kunna upptäckas (se Sjöbergh 2005)

SnålGranska: Maskininlärning för feldetektion

- Se problemet som ett taggningsproblem
- Märk upp felen t.ex. med taggen ERROR och resten med OK
- Träna en maskininlärningsalgoritm (t.ex. TBL) på det annoterade materialet + material med korrekt text.
- Utvärdera på ett okänt men annoterat material.

SnålGranska

Vi behöver många fel

Idé: Skapa fel automatiskt

Träna en maskininlärningsalgoritm på detta material. Automatisk annotering. Förlåtande felgenerering.

Skapa en “feltaggare” för varje feltyp

Skapa träningsdata

1. Ta en referenskorpus
2. Ta en kopia av denna referenskorpus.
3. Skapa särskrivningsfel i denna – märk upp dessa ERROR
4. Övriga ord märks upp med OK i de två korpusarna.

Familjen	NN	OK
bodde	VB	OK
i	PP	OK
ett	DT	OK
hus	NN	OK
och	KN	OK
körde	VB	OK
en	DT	OK
miljövänlig	JJ	OK
bil	NN	OK
.	MAD	OK

Kvinnan	NN	OK
hade	VB	OK
köpt	VB	OK
en	DT	OK
ny	JJ	OK
hus	NN	ERROR
bil	NN	ERROR
.	MAD	OK

SnålGranska

Fördelar:

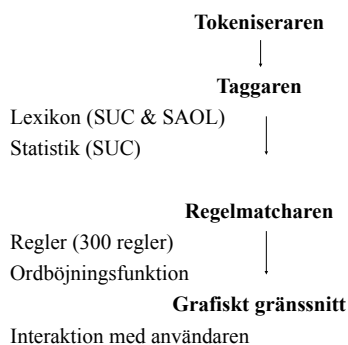
- Begränsad manuell insats
- En taggare per feltyp kan ge bra diagnos och även ersättningsförslag (åtminstone för sårskrivningar).

Nackdelar:

- Varje feltaggares enskilda falsklarm kan resultera i många falsklarm om man sätter ihop dem i ett verktyg.

Program	Detektion	Diagnos	Förslag
Critique	Ja	Ja	Ja
Granska	Ja	Ja	Ja
Grammatifix	Ja	Ja	Ja
Scarrie	Ja	Ja	Nej
FiniteCheck	Ja	Ja	Nej
ProbGranska	Ja	Nej	Nej
SnålGranska	Ja	Ja	Nej*
Confusion sets	Ja	Ja	Ja

Granskas uppbyggnad



Första av allt: Namngivning av regelkategori och regler

```

category regler {
  info("exempelregler")
  link("www" "namn på länk")
}

regelnamn@regler
{
  x()
  -->
  action(scrutinizing)
}
  
```

```

Kvinnan nn.utr.sin.def.nom
hade vb.prt.akt.aux
köpt vb.sup.akt
en dt.utr.sin.ind
ny jj.pos.utr.sin.ind.nom
hus nn.neu.plu.ind.nom
bil nn.utr.sin.ind.nom

```

```

ex2@regler
{
X(wordcl=nn & spec=ind &
case=nom), % hus
Y(wordcl=nn) % bil
-->
mark(X Y)
corr(X.join(Y.text))
action(scrutinizing)
}

```

```

ex1@regler
{
X(text="ett"),
Y(wordcl=nn & gender=utr)
-->
mark(X Y)
corr(X.form(gender:=Y.gender))
action(scrutinizing)
}

```

```

ex2@regler
{
X(text="gå"),
Y(text="över"),
Z(text="nån"),
Z2(text="efter"),
Z3(text="vatten")
-->
mark(Z)
corr(Z.replace("ån"))
info("Misstänkt fel: det heter nog gå över
ån efter vatten")
}

```

*en elakt hund
ett liten hus*

```

ex4@regler {
X1(wordcl=dt),
X2(wordcl=jj),
X3(wordcl=nn & (gender!=X2.gender | num!
=X2.num | spec!=X2.spec) &
(gender=X1.gender & num=X1.num &
spec=X1.spec))
-->
action (scrutinizing)
corr (X2.form(gender:=X3.gender,
num:=X3.num, spec:=X3.spec)) }

```

Generering av ersättningsförslag

Här ligger några trevligt restauranger:

Problem: Om varje ord i lexikonet kopplas explicit till alla dess böjningsformer blir lexikonet väldigt stort.

Lösning: Basformen *trevlig* slås upp i lexikonet.

Varje basform i lexikonet har en böjningsregel.

Med hjälp av böjningsregeln för *trevlig* hittas den rätta formen *trevliga*.

Anropas med metoden

x.form(x.särdrag:=särdragsvärde)

Varning: Granska granskar sig själv!

bil --> bil r1
hus --> hus r5

	SIN IND NOM	SIN IND GEN	SIN DEF NOM	SIN DEF GEN	PLU IND NOM	PLU IND GEN	PLU DEF NOM	PLU DEF GEN
1	”	s	en	ens	ar	ars	arna	arnas
5	”	”	et	ets	”	”	en	ens

37

Viktiga metoder för ersättningsförslag

insert (Y) tar Y som text, t.ex. Y.text och sätter in det före det angivna objektet, X.insert(Y)

join (Y) tar Y som text enligt: X.join(Y.text) och sätter ihop dem till en textsträng (ord)

delete tar bort det angivna objektet, enligt: X.delete().

replace(Y) tar Y som text, Y.text, och ersätter det angivna objektet med Y, enligt X.replace(Y.text)

Taggning av felaktiga konstruktioner

Här taggas hylla som nomen:
Vi har köpt en bok hylla sa Per.

Här taggas hylla som verb:
Vi har köpt en bok hylla på IKEA sa Per.

Anpassa sig till taggaren? Tillgång till fler tolkningar av ordet + lexikal sannolikhet

Hur man skriver regler för att hitta felaktiga särskrivningar

I grova drag för t.ex. *språk granskning* eller *signal detektionsteori*:

1. Leta upp två substantiv i obestämd form som står i följd.

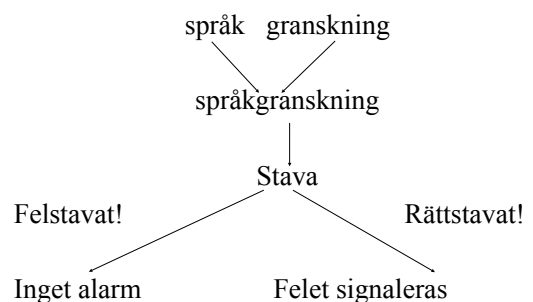
Korrekta detektioner: I grova drag för t.ex. **språk** **granskning** eller **signal** **detektionsteori**

Falska alarm: Jag kan tänka mig att en **massa** **bedömare** har gjort en **rad** **studier** i ämnet.

2. Ta bort så många falska alarm som möjligt.
Ställ ytterligare krav på matchningen.

Viktigaste kravet: det första substantivet får inte tillhöra gruppen substantiv som kan utgöra en måttapposition, t.ex. *grupp, antal, rad, massa*.

3. Använd Stava



Särskrivningsregel

```
ex5@regler {  
X(wordcl=nn & spec=ind & case=nom &  
  nntype!= set & nntype!=dat),  
Y(wordcl=nn & spec=ind &  
  spell_OK(smart_concat(X.text,  
    Y.text), X.token))  
-->  
mark(X Y)  
corr(X.join(Y.text))  
info("Felaktig särskrivning")  
action(scrutinizing)  
}
```

Uppgift: skriv en regel som
upptäcker och åtgärdar formfel
som i: *Jag har spela grammofon.*

Jobba gärna i par

Fråga!

Lösning?

Några resultat vid utvärderingen av Granska

Begränsad undersökning

I populärvetenskaplig text upptäcktes 9 av 10
fel, 5 av 10 felrapporter var korrekta

Gymn.- och högskoletexter upptäcktes 4 av 10
fel, 7 av 10 felrapporter var korrekta

Användaranpassning av Granska

- Andraspråksinlärare och användare med speciella behov
- Andra behov av språklig hjälp
- Om en mening innehåller många fel är den synnerligen svår att analysera
- Nya krav på gränssnitt och innehåll
Grim!