Sats-segmentering för svenska

Andreas Brandt
University of Lund
dat01anb@ludat.lth.se

Abstract

In this report I describe a method for segmenting POS-tagged swedish discourse into clauses.

I have implemented this method for two different tag-sets, and evaluated the results.

1 Introduktion

Att dela upp en text i satser är inte något som man ofta har som slutmål i en applikation, men ofta har man stor användning av det då man behöver analysera semantiken i en text. I stället för att analysera meningar, som ofta har flera semantiska betydelser, är det lättare att analysera segment som var för sig har en innebörd. Användningsområdena är främst andra lingvistiska applikationer, så som t.ex. talsyntes, röstigenkänning, textanalys, maskinöversättning och kunskapsinsamling. Det är tänkt att Carsim[1] ska använda denna metod för att extrahera information ur text, och konvertera det till scener.

Styrkan hos den här metoden är att den är robust eftersom den inte kräver någon generering av parse-träd och liknande, utan bara en genomgång av texten.

2 Bakgrund

En sats definieras här utifrån hur vi uppfattar en uppläst text. I studier utförda av (E. Strangert, 1993) använder man akustisk information, i form av tysta intervall och tonfall, och perceptuellt uppfattade pauser för att placera sats-avskiljare. Det visar sig att alla dessas, samt den algoritmiska metodens(Ejerhed, 1990), avskiljare inte är oberoende. Utan tvärtom, den algoritmiska metodens avskiljare sammanfaller nästan alltid med någon av de andra. Detta visar att metoden verkar fungera.

3 Metod

Indata till sats-segmenteraren är en POS-taggad text, med taggar enligt SUC[2].

Texten gås igenom och varje bigram, trigram eller 4-gram, kontrolleras mot en uppsättning av regler(Ejerehed, 1996)(se Appendix). Reglerna talar om var en sats-avgränsare ska sättas in.

4 Implementation

Implementeringen skedde i Java, med dess paket för reguljära uttryck[3]. Implementationen stöder både SUC's och Granska's[4] tagg-uppsättning.

4.1 SUC

Det finns två olika format på SUC, jag valde SUC1A. Det kan se ut så här:

```
("<Vi>" <86>
        (PN UTR PLU DEF SUB "vi"))
("<sitter>"     <87>
        (VB PRS AKT "sitta"))
("<här>"        <88>
        (AB "här"))
("<i>"  <89>
        (PP "i"))
("<solen>"      <90>
        (NN UTR SIN DEF NOM "sol"))
("<tills>"      <91>
        (SN "tills"))
("<det>"        <92>
        (PN NEU SIN DEF SUB/OBJ "det"))
("<är>" <93>
```

[1] http://www.df.lth.se/ richardj/carsim/

[2] Stockholm Umeå Corpus

[3] java.util.regex.*

[4] En POS-taggare av Johan Carlberger

```
        (VB PRS AKT "vara"))
("<dags>"       <94>
        (NN UTR - - - "dags"))
("<.>"  <95>
        (DL MAD "."))
```

Vi plockar ut varje symbol(token), tagg och lemma och stoppar in i varsin array. Resultatet blir att symbol[n] har tagg[n] och lemma[n]. Därefter blir det mycket enkelt att kontrollera bi-gram, tri-gram och 4-gram. Vi har även en array av samma storlek som de övriga, med sats-avgränsare. Finner vi att det enligt reglerna ska vara en sats-avgränsare på position n, så sätter c[n] = sant. Därefter har man all nödvändig information.

Exempel på utdata:

```
<c> Den kallas allmänt stenbit , </c>
<c> men det är bara hannarna </c>
<c> som heter så </c>
```

4.2 Granska

Jag har även implenterat en segmenterare för Granska. Granska har en annan uppsättning av taggar, men så gott som alla nödvändiga kategorier finns med. Det saknas taggar för paragrafer och överskrifter, men dessa spelar en underordnad roll. En text taggad med Granska kan se ut så här:

```
Vi      pn.utr.plu.def.sub      vi
sitter  vb.prs.akt      sitta
här     ab      här
i       pp      i
solen   nn.utr.sin.def.nom      sol
tills   sn      tills
det     pn.neu.sin.def.sub/obj  det
är      vb.prs.akt.kop  vara
dags    nn.utr  dags
.       mad     .
```

5 Utvärdering

Sats-segmenteringen fungerar förvånansvärt bra, och många av feltaggningarna tycks kunna gå att kringå med exempelvis specialfall. En del fel beror också på hur texten är POS-taggad, inte bara fel-taggning, utan också vilken policy som använts.

I (Ejerehed, 1996) uppges att metoden ger 96% korekt taggning på SUC. Detta ligger nog ganska nära sanningen, jag har kontrollerat ett textstycke med 98 funna satser. Jag hittade tre fall som taggats fel, vilket ger ungefär samma siffra. Att finna fel är inte helt trivialt, då man oftast får göra en mer eller mindre subjektiv bedömning.

5.1 Feltaggning

Enligt specialfallet i regel 4(se Appendix) ska en brytning ske här:

```
<c>
Vi      (PN UTR PLU DEF SUB)
träffas         (VB PRS SFO)
</c>

<c>
där     (AB)
och     (KN)
sen     (AB)
tar     (VB PRS AKT)
vi      (PN UTR PLU DEF SUB)
oss     (PN UTR PLU DEF OBJ)
hit     (AB)
till    (PP)
.
.
.
```

Här borde en brytning skett:

```
<c>
Om      (SN)
vi      (PN UTR PLU DEF SUB)
inte    (AB)
ses     (VB INF SFO)
så      (AB)         <---
ses     (VB INF SFO)
vi      (PN UTR PLU DEF SUB)
om      (PP)
tre     (RG NOM)
månader         (NN UTR PLU IND NOM)
här     (AB)
.       (DL MAD)
</c>
```

5.2 Gränsfall

Ett ganska vanligt förekommande fall av ett tveksamt fel, är när hjälpverbet “ha” förekommer. Här kan man tycka att ingen satsbrytningen skulle ha skett.

```
<c> Jag har </c>
<c> fått pengar </c>
```

6 Förslag

Eftersom SUC har en reltaivt fattig uppsättning av taggar, kan man säkerligen hitta fler regler med en större uppsättning. Exempelvis fattas taggning för olika former av hjälpverb i SUC. Med t.ex. Granskas mer nyansrika taggar är det kanske möjligt att skriva regler för vissa fall som inte kan fångas upp annars.

7 Slutsats

Sats-segmentering med sats-avgränsare istället för metoder som använder sig av nästlade satser(sats, bi-sats), är en metod som är både enkel att implementera och effektiv att använda.

Det verkar även finnas bra möjligheter att förbättra metoden om man har nyansrikare taggar, som i Granska.

Acknowledgements

Tack till Richard Johansson för hjälp och tips.

References

D. Huber E. Strangert, E. Ejerhed. 1993. Clause structure and prosodic segmentation. *Papers from the Seventh Swedish Phonetics Conference*.

E. Ejerehed. 1996. Finite state segmentation of discourse into clauses. *Extended Finite State Models of Language*.

E. Ejerhed. 1990. A swedish clause grammar and its implementation. *The 7th Nordic Conference on Computational Linguistics*.

Appendix

A Stycket som utvärderades

```
<c> Jag vet inte . </c>
<c> Du får sova hos mig , </c>
<c> sa Lena så tyst </c>
<c> att bara Torvald hörde det . </c>
<c> Men det kan inte bli som förra
gången . </c>
<c> Jag lovade Matt . </c>
<c> Jag förstår , </c>
<c> sa Torvald . </c>
<c> Min ryggsäck står där inne under
flipperspelet , </c>
<c> sa Gunnar . </c>
<c> Hur full jag än är </c>
<c> måste jag och Rosita i väg .</c>
<c> Det ordnar sig , </c>
<c> sa Torvald . </c>
<c> När kommer Rosita hit . </c>
<c> Jag vet inte , </c>
<c> sa Gunnar . </c>
<c> Men hon kommer . </c>
<c> Hon måste ju ha </c>
<c> förstått att snuten höll mig kvar .</c>
--<c> Jag har </c>
<c> suttit hos snuten i natt , </c>
<c> sa han sen förklarande till alla </c>
--<c> som eventuellt inte hade </c>
<c> hört det . </c>
<c> I det ögonblicket stannade Bill Marshall
sin Morgan utanför Monaco . </c>
<c> Torvald , </c>
<c> sa han . </c>
<c> Kan vi åka en sväng . </c>
<c> Jag måste snacka med dig . </c>
<c> Torvald visste inte mycket om bilar men
en grön Morgan är en grön Morgan </c>
<c> och ser ut som en bil . </c>
<c> Torvald reste sig . </c>
<c> Vi ses på Gaudeamus , </c>
<c> sa han . </c>
<c> Vid lunch . </c>
<c> Om jag inte kommer så </c>
<c> ta mina pengar , Gunnar , </c>
<c> och gör rätt för oss </c>
<c> och bjud på lunch . </c>
<c> Men jag kommer . </c>
<c> Annars är jag på Select
klockan fyra . </c>
<c> Sen fick han en idé . </c>
<c> Han gick fram och </c>
<c> snackade med Bill Marshall </c>
<c> och kom sen tillbaka . </c>
<c> Han sa till Gunnar : </c>
<c> Bill kör dig och Rosita Zoraffie
till stationen . </c>
<c> Han hämtar er här halv sex . </c>
<c> Om vi inte ses , Gunnar , </c>
<c> vilket vi gör , </c>
<c> men om vi inte ses så </c>
<c> skriver jag till dig . </c>
<c> Om några månader är det jobb i
```

```
Schweiz på tretusen
meters höjd över Sion . </c>
<c> Vi anmäler oss dit . </c>
--<c> Vi träffas </c>
<c> där och sen tar vi oss hit till
Monaco igen för en månadsfylla </c>
<c> och sen sticker vi hem . </c>
<c> Hälsa nu alla . </c>
<c> Rör inga kalabriska flickor . </c>
<c> Krama om Peppino och
den döva killen . </c>
<c> Jag avundas dig lite . </c>
<c> Behåll pengar </c>
<c> så ni åtminstone kommer lite
söder om Neapel . </c>
<c> Det som är kvar </c>
<c> kan du ge till Bibi . </c>
<c> Okey , </c>
<c> sa Gunnar . </c>
<c> Om vi inte ses så ses vi om
tre månader här . </c>
<c> Jag ska forska i schweizjobbet . </c>
<c> Torvald klev in i Bill Marshalls </c>
<c> Morgan och Bill sa : </c>
<c> Vi sticker till ett ställe </c>
<c> där vi får vara i fred . </c>
<c> Vad tror du om kallt vitt vin , </c>
<c> sa Torvald . </c>
<c> Det tror jag på , </c>
<c> sa Bill . </c>
<c> Kör till Saint-Sulpice , </c>
<c> sa Torvald . </c>
<c> Jag ska visa dig ett nytt ställe </c>
<c> där vi får vara i fred . </c>
<c> Dig har jag </c>
<c> önskat träffa </c>
<c> sen livet upphörde för nån dag sen .</c>
<c> I lördags . </c>
<c> De for iväg </c>
<c> och Bill parkerade bilen i närheten
av den rostfria baren </c>
<c> och sen gick de in där . </c>
<c> De möttes av en glad och pigg Jerry </c>
<c> som omedelbart presenterade sig
för Bill Marshall </c>
<c> och hämtade en flaska av sitt vin .</c>
<c> Han hämtade två glas och en askkopp .</c>
<c> Han slog i glasen </c>
<c> och sen lät han dem vara i fred . </c>
<c> Har du letat efter mig , </c>
<c> sa Bill . </c>
<c> Ja , </c>
<c> sa Torvald . </c>
```

B

B.1 Regler(Ejerehed, 1996)

```
1 PUNCTUATION
1a
<h> XX => <h> <c> XX
<p> XX => <p> <c> XX

1b
DL MAD XX => DL MAD <c> XX,
where XX is not end tag

1c
DL MID FIN => DL MID <c> FIN
1d
DL MID XX FIN => DL MID <c> XX FIN,
where XX= PN, NN, PM or AB


2 COMPLEMENTIZERS

2a
special:
XX KN SN => XX <c> KN SN
general:
XX SN => XX <c> SN

2b
special:
XX KN HX => XX <c> KN HX
XX HX HX => XX <c> HX HX
general:
XX HX => XX <c> HX


3 KN+FINITE VERB

special:
XX KN FIN => <c> XX KN FIN,
where XX is a closed class of
finite forms of the verbs
''vara'', ''gå'', ''stå'', ''sitta''
general:
XX KN FIN => XX <c> KN FIN$

4 KN+XX+FINITE VERB,
where XX=PN, NN, PM or AB

special:
YY KN XX FIN => <c> YY KN XX FIN,
 if YY=XX
general:
YY KN XX FIN => YY <c> KN XX FIN,
 if YY<=XX


5 SEQUENCES OF FINITE VERBS

5a CASE: 0 WORDS BETWEEN FINITE VERBS
FIN FIN => FIN <c> FIN

5b CASE: 1 WORD BETWEEN FINITE VERBS
FIN XX FIN => FIN XX <c> FIN
```

```
5c CASE: 2 WORDS BETWEEN FINITE VERBS
special:
FIN YY XX FIN => FIN YY <c> XX FIN,
where XX=PN, NN, or PM
general:
FIN YY XX FIN => FIN YY XX <c> FIN
```

B.2 Taggar(Ejerehed, 1996)

```
<h>
 head

<p>
 paragraph

<<<<kk09>>>>
 block

</h>
 end head

</p>
 end paragraph

<<<<</kk09>>>>
 end block

DL MAD
 major delimiter ( . ? ! )

DL MID
 minor delimiter ( ,  : )

FIN
 VB PRS AKT, VB PRS SFO,
 VB PRT AKT, VB PRT SFO,
 VB SUP AKT, VB SUP SFO,
 VB IMP AKT

PN
 PN ... SUB, PN ... SUB/OBJ
 (subject forms of pronouns)

NN
 NN ... NOM

PM
 PM NOM

AB
 AB, AB POS, AB KOM, AB SUV
 (adverbs)

KN
 conjunction

SN
 subjunction

HX
 HA, HD..., HP..., HS...
 (Wh: adverbs, determiners,
  pronouns, possessives)
```