Generating a new Narnia

Tim Stolp

Universiteit van Amsterdam

Inleiding

In dit project wordt er gekeken naar taal. Hoe zit een taal in elkaar en kunnen er grammaticale zinnen automatisch gegenereerd worden. Om hierachter te komen wordt er een corpus ontleed en de regels van de zinnen in een Context-Free Grammar (CFG) gezet en hiermee zou er kunnen worden voorspeld welke woorden na elkaar komen en zo automatisch zinnen genereren. Een CFG bevat grammaticale regels die zeggen welke soort woorden na elkaar kunnen voorkomen en welke woorden er per woordsoort gekozen kunnen worden. Als corpus is er gekozen voor het boek narnia. Narnia is wereldberoemd en een bestseller. Kan een goed boek worden gegenereerd worden door middel van een CFG? Met genoeg data zouden er goede verhalen kunnen worden gemaakt maar aangezien er hier nog niet met context wordt gewerkt en vooral met waarschijnlijkheid van een opvolgend woord is de kans klein dat er een origineel goed lopend verhaal kan worden gemaakt.

Methode

De corpus die er hier wordt gebruikt zijn de eerste 1000 woorden van het boek Narnia. Interpunctie is in de corpus weggelaten. Deze zinnen moeten worden ontleed door middel van *constituency parsing*. Bij *constituency parsing* wordt er gebruik gemaakt van een CFG waarmee een zin wordt opgedeeld in *constituents*. Deze *constituents* combineren samen tot andere *constituents* met als uiteindelijke doel om de *constituents* te kunnen combineren tot S (sentence) door middel van regels in de CFG. Het is belangrijk om zorgen dat er zo min mogelijk *ambiguity* is. Er is *ambiguity* als een zin op meerdere manieren geparsed kan worden. De zinnen worden handmatig geparsed en de gevonden regels worden aan de CFG toegevoegd. Als een zin te lastig is om hiermee te ontleden dan wordt deze herschreven tot een makkelijkere zin met dezelfde betekenis. Met de uiteindelijke CFG worden er 50 nieuwe zinnen gemaakt en gekeken of deze correct zijn.

Resultaten

Met de CFG kunnen de eerste paar regels geparsed worden. Deze regels zouden voor een groot deel de corpus moeten kunnen *parsen* nadat de lexicon van de andere zinnen is toegevoegd.

```
# sentences
  S -> NP VP
  S -> VP
  S -> S WHNP
  S -> S CC S
  SBAR -> WHNP VP
```

```
# constituents
   NP -> Det N
   NP -> Det N N
   NP -> N N
   NP -> NNP
   NP -> Adj N
   NP -> Det ADJP N
   NP -> OP NP
   NP -> Pro CC Pro
   NP -> Pro
   NP -> NN SBAR
   NP -> Adv

   VP -> V NP
   VP -> V ADVP PP
   VP -> V ADJP PP
   VP -> V PP
   VP -> VBD VP

   PP -> TO NP
   PP -> IN NP
   PP -> IN IN NP
   PP -> Prep NP
   PP -> PP PP

   ADVP -> Adv

   ADJP -> Adj N
   ADJP -> Adj
   ADJP -> ADJP Comma ADJP

   WHNP -> Pos N VP
   WHNP -> Det

   OP -> Pro Comma

   # lexicon
   Det ->  'the' | 'an' | 'this' | 'that'
   V -> 'were' | 'sent' | 'is' | 'happened' | 'lived'
   VBD -> 'were'
   Adj -> 'four' | 'old' | 'ten' | 'nearest' | 'two'
   Adv -> 'away' | 'there'
   IN -> 'from' | 'during' | 'because' | 'of' | 'in'
   TO -> 'to'
   N -> 'children' | 'names' | 'war' | 'air-raids' | 'house' | 'professor' \
        | 'story' | 'heart' | 'country' | 'post' | 'office' | 'railway' | 'station' | 'miles' \
```

NNP -> 'london'
NN -> 'something'
Pos -> 'whose'
Pro -> 'peter' | 'susan' | 'edmund' | 'lucy' | 'they' | 'them' | 'he'
CC -> 'and'
Comma -> ','
Prep -> 'about'

Discussie

In de corpus zitten veel zinnen waarin een personage iets zegt. Dit soort zinnen waren lastig om goed te *parsen* met de CFG aangezien uitgesproken taal vaak veel bijzinnen en spreektaal gebruikt waarin dingen worden weggelaten voor het gemak van de spreker. Dit zorgt voor incomplete zinnen die zorgen voor regels in de CFG die problemen kunnen veroorzaken zoals *ambiguity* en gegenereerde zinnen die niet correct engels zijn. Om deze zinnen toch te kunnen *parsen* zijn er regels die zeggen dat bepaalde *constituents* gevolgd kunnen worden door sub-zinnen of gehele nieuwe zinnen.
Om lange opsommingen te kunnen *parsen* zitten er recursieve regels in de CFG. Bij het genereren van nieuwe zinnen komt het in een loop te zitten waar er telkens hetzelfde woord aan de opsomming wordt toegevoegd. Dit kon deels verholpen worden door de recursieve regels lager in de CFG te zetten waardoor de andere regels eerst worden afgegaan.