Oppgave 5, Syntaks og klassifisering 24

October 21, 2024

1 Strukturell flertydighet

Anta følgende grammatikk for et fragment av norsk:

$S \rightarrow NP\ VP$

$NP \rightarrow D\ N$

$NP \rightarrow NP\ PP$

$VP \rightarrow V \mid V\ NP$

$VP \rightarrow VP\ PP$

$PP \rightarrow P\ NP$

$D \rightarrow$ en|et

$N \rightarrow$ barn|baby|vindu|vugge

$V \rightarrow$ sover

$P \rightarrow$ i | ved

1. Hvor mange analyser tildeler grammatikken til setningene under?

(a) en baby sover i en vugge ved et vindu

(b) et barn i en vugge sover

2. Tegn treet eller trærne grammatikken tilordner setning **(a)**.

3. Utvid grammatikken slik at den tillater vilkårlig mange adjektiv foran substantivet i substantivfrasen og godtar setninger som: "*en liten søt baby sover i en stor vugge*"

4. Hvor mange analyser tilordner den utvidede grammatikken din til setningen "*en rund liten søt baby sover*"?

2 En grammatikk for norsk

NLTK inneholder flere forskjellige parsere som automatisk tildeler syntaktisk struktur til en setning, i henhold til en grammatikk. Her skal du bruke `RecursiveDescentParser` som står beskrevet i kapittel 8, del 3 i NLTK-boka. Du kan formulere grammatikken din direkte som en streng over flere linjer, slik:

[3]:
```
import nltk

grammar = nltk.CFG.fromstring('''
  S -> NP VP
  VP -> V NP | V NP PP
  PP -> P NP
  V -> "saw" | "ate" | "walked"
  NP -> "John" | "Mary" | "Bob" | Det N | Det N PP
  Det -> "a" | "an" | "the" | "my"
  N -> "man" | "dog" | "cat" | "telescope" | "park"
  P -> "in" | "on" | "by" | "with"
  ''')
```

Pass på at du skriver pilene som `->`, ikke → eller `→`.

`grammar` er her et objekt av typen `nltk.CFG.grammar` returnert av metoden `CFG.fromstring()` som finnes i NLTK. Forkortelsen CFG står for "context free grammar".

Merk at `RecursiveDescentParser` som nå skal anvende grammatikken, ikke håndterer venstre-rekursjon, av typen `VP → VP PP`, så du må formulere grammatikken uten denne formen for rekursjon.

`RecursiveDescentParser` tar imot en tokenisert setning. Det vil si en liste med strenger. Her bruker vi bare `split()` til å tokenisere fordi vi ikke har noen tegnsetting i setningene.

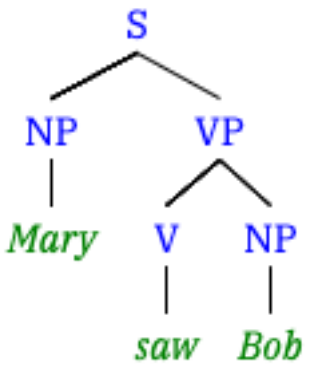
[5]:
```
# Tokenisert setning
sent = "Mary saw Bob".split()

# Sender grammatikken inn og får en parser i retur
parser = nltk.RecursiveDescentParser(grammar)

# Skriver ut hvert tre (her bare ett)
for tree in parser.parse(sent):
    print(tree)
```

```
(S (NP Mary) (VP (V saw) (NP Bob)))
```

Utskriften tilsvarer det syntaktiske treet under.



Slike grafiske trær kan genereres med samtlige nettverktøy eller tegnes manuelt med et tegneprogram. *Bruken av disse nettverktøyene inngår ikke i pensum. Vi anbefaler at du øver på å tegne syntaktiske trær på egen hånd.*

1.

Du skal nå implementere en kontekstfri grammatikk med denne parseren i Python som analyserer et fragment av norsk, slik at setningene under gis riktig analyse:

(a) (S (NP Per) (VP (V gir) (NP (Det en) (N bok)) (PP (P til) (NP Kari))))

(b) (S (NP Kari) (VP (V gir) (NP Per) (NP boka)))

(c) (S (NP Ola) (VP (V sover)))

(d) (S (NP Kari) (VP (V spiser)))

(e) (S (NP Kari) (VP (V spiser) (NP middag)))

(f) (S (NP Per) (VP (V finner) (NP boka)))

2.

Grammatikken, slik den er implementert, er imidlertid ikke tilfredsstillende, siden den for eksempel vil godta setninger som "*Kari sover boka*" og "*Ola finner*". Verifiser dette ved å skrive ut analysene som grammatikken din tildeler disse setningene.

3.

Du skal nå skrive en ny og forbedret versjon av grammatikken slik at de grammatiske konstruksjonene i **1.** tillates, men ugrammatiske konstruksjoner (som "*Kari sover boka*" og "*Ola finner*") er utelukket. Skriv ut analysene som den nye og forbedrede grammatikken tildeler de grammatiske setningene i **1.**, og vis videre at de ugrammatiske setningene ikke tildeles noen analyse.

3 Naive Bayes-klassifisering

I oppgavene under skal vi bruke Naive bayes-formelen:

$$\hat{k} = \underset{k \in K}{\text{argmax}} P(k) \prod_{j=1}^{n} P(v_j|k)$$

Den estimerte kategorien $\hat{k}$ for et dokument er lik den kategorien som får høyest verdi (argmax$_{k \in K}$) når vi gjør følgende:

Vi ganger sannsynligheten for kategorien $P(k)$ med produktet av sannsynlighetene $\prod_{j=1}^{n} P(v_j|k)$ for alle ordene i dokumentet vi vil klassifisere.

Vi beregner sannsynlighetene som følger:

Sannsynligheten for en kategori er lik antall ganger kategorien forekommer i korpuset, delt på antall kategorier totalt. Sannsynligheten for et ord er lik antall ganger ordet forekommer i en gitt kategori, delt på antall token i den kategorien.

Hvis vi bruker legg-til-én-glatting, legger vi til én over brøkstreken, og legger til lengden av hele ordforrådet (antall ordtyper). Merk at et dokument kan være så kort eller langt vi vil, selv kun én setning eller noen få ord.

3.1 Sannsynlighet for en setning i en kategori

Under har vi et lite korpus med filmanmeldelser, og vil regne ut sannsynligheten for at en ny setning er del av en positiv anmeldelse eller en negativ anmeldelse. Under er sannsynlighetene for hvert ord gitt kategorien *positiv* eller *negativ*. Sannsynlighetene for ordene er allerede regnet ut. Vi antar at det er like mange positive klasser som negative, altså er sannsynligheten for hver av klassene lik.

Ord	POS	NEG
jeg	0,09	0,16
liker	0,07	0,06
alltid	0,29	0,06
utenlandske	0,04	0,15
filmer	0,08	0,11

Hvilken klasse vil Naive Bayes gi setningen "*Jeg liker alltid utenlandske filmer*"?

3.2 Sannsynlighet for at et dokument er i en kategori

I denne oppgaven har vi noen dokumenter med ord, som tilhører sjangerne `action` eller `komedie`. Bruk Naive Bayes-formelen til å klassifisere det nye dokumentet.

Dokument	Kategori
moro, par, kjærlighet, kjærlighet	komedie
rask, rasende, skyte	action
par, fly, rask, moro, moro	komedie
rasende, skyte, skyte, moro	action
fly, rask, skyte, kjærlighet	action

Dokumentet vi vil klassifisere, inneholder følgende ord: **rask**, **par**, **skyte**, **fly**

Du må altså regne ut sannsynlighetene for de forskjellige ordene selv, men du trenger bare å regne ut for ordene i det nye dokumentet. **Bruk legg-til-én-glatting (Laplace)**. Regn så ut hvilken verdi som er størst. Blir dokumentet klassifisert som `action` eller `komedie`?