IN2110 - Obligatorisk oppgave 2a

Emma Dae, Morten Storvik
Mark Tzvetoslavov

April 26, 2021

1 Dependesgramatikk

1.1 Dependsgrafer

Det vi har gjort i følgende oppgave er å tegne grafen for setningen i tabell 2. Setningen er annotert med pos-tagger og dependensralsjoner.

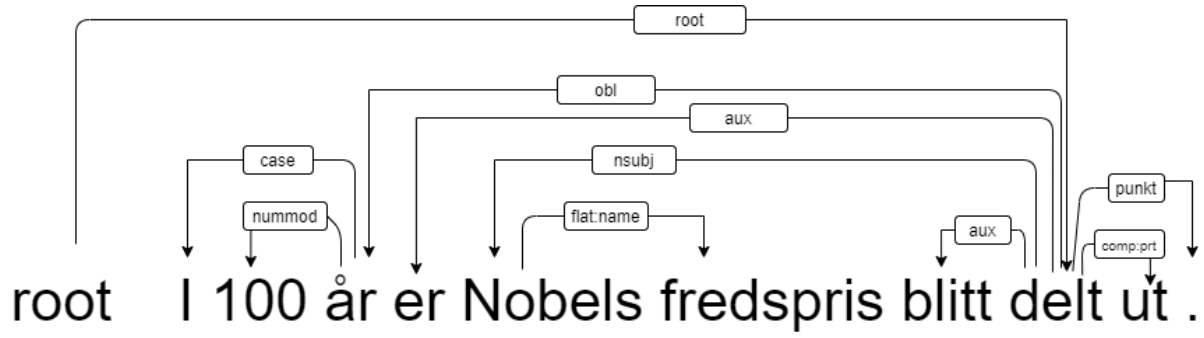

Figure 1: grafen for setningen - ”I 100 år er Nobels Fredspris blitt delt ut.”

1.2 Transisjonsparsing

For denne delen av oppgaven, skal vi vise en full sekvens av transisjonsoperasjoner for depensgrafen fra forrige deloppgave. Vi har valgt å gå for ”arc eager-algoritmen”.

Begge algoritmene har tre transisjonsoperasjoner til felles:

- SHIFT
- LEFT-ARC
- RIGHT-ARC.

Arc eager har i tillegg til disse REDUCE, som tar ut ordet på toppen av stacken, men krever at du har et eksisterende head.

steg	stack	ordliste	operasjon	Relasjon lagt til
0	[root]	[I, 100, år, er, Nobels, Fredspris, blitt, delt, ut, .]	SHIFT	
1	[root, I]	[100, år, er, Nobels, Fredspris, blitt, delt, ut, .]	SHIFT	
2	[root, I, 100]	[år, er, Nobels, Fredspris, blitt, delt, ut, .]	SHIFT	
3	[root, I, ~~100~~, år]	[er, Nobels, Fredspris, blitt, delt, ut, .]	LEFT-ARC	(100 <- år) nummod
4	[root, ~~I~~, ~~100~~, år]	[er, Nobels, Fredspris, blitt, delt, ut, .]	LEFT-ARC	(I <- år) case
5	[root, ~~I~~, ~~100~~, år, er]	[Nobels, Fredspris, blitt, delt, ut, .]	SHIFT	
6	[root, ~~I~~, ~~100~~, år, er, Nobels]	[Fredspris, blitt, delt, ut, .]	SHIFT	
7	[root, ~~I~~, ~~100~~, år, er, Nobels, Fredspris]	[blitt, delt, ut, .]	SHIFT	
8	[root, ~~I~~, ~~100~~, år, er, Nobels, ~~Fredspris~~]	[blitt, delt, ut, .]	RIGHT-ARC	(Nobels -> Fredspris) flat
9	[root, ~~I~~, ~~100~~, år, er, Nobels,]	[blitt, delt, ut, .]	REDUCE	
10	[root, ~~I~~, ~~100~~, år, er, Nobels, blitt]	[delt, ut, .]	SHIFT	
11	[root, ~~I~~, ~~100~~, år, er, Nobels, blitt, delt]	[ut, .]	SHIFT	

Figure 2: første del til sekvensen

12	[root, ~~I~~, ~~100~~, år, er, Nobels, ~~blitt~~, delt]	[ut, .]	LEFT-ARC	(blitt <- delt) aux
13	[root, ~~I~~, ~~100~~, år, er, ~~Nobels~~, ~~blitt~~, delt]	[ut, .]	LEFT-ARC	(Nobels <- delt) aux
14	[root, ~~I~~, ~~100~~, år, ~~er~~, ~~Nobels~~, ~~blitt~~, delt]	[ut, .]	LEFT-ARC	(er <- delt) aux
15	[root, ~~I~~, ~~100~~, ~~år~~, ~~er~~, ~~Nobels~~, ~~blitt~~, delt]	[ut, .]	LEFT-ARC	(år <- delt) obl

Figure 3: andre del til sekvensen

16	[root, I, 100, år, er, Nobels, blitt, delt, ut]	[.]	SHIFT	
17	[root, I, 100, år, er, Nobels, blitt, delt, ut, .]	[]	SHIFT	
18	[root, I, 100, år, er, Nobels, blitt, delt, ut, .]	[]	RIGHT-ARC	(delt -> ut) comp:prt
19	[root, I, 100, år, er, Nobels, blitt, delt, ut, .]	[]	RIGHT-ARC	(delt -> .) punkt
20	[root, I, 100, år, er, Nobels, blitt, delt, ut, .]	[]	RIGHT-ARC	(root -> delt) .

Figure 4: tredje del til sekvensen

2 Trene modeller

I denne delen av obligen har vi brukt den norske bokmåls-trebanken fra Universal Depencies. Dataene kommer i CoNLL-U-format og måtte konverteres til spaCy sitt json-format før vi kunne trene modellen.Vi bare tok i bruk følgende kode-linje for å trene den:

```
python -m spacy train -p parser nb modellmappe no_bokmaal-ud-train.json no_bokmaal-ud-dev.json
```

3 Evaluering

Det vi skal gjøre i denne delen er å evaluere parseren. Vi skal parse development datasettet no-bokmaal-ud-dev.conllu med parseren vi har trent. Den ferdig trente modellen fra forrige oppgave lastet vi inn i spacy:

```
import spacy
nb = spacy.load("my-model/model-best")
```

3.1 Attachment score

I følgende del av obligen fullførte vi funksjonen attachmentScore(). Den tar inn to argumenter, nemlig: en liste med "gull"-setninger og en liste med prediksjoner. Den ender opp med å returnere en tuppel med to tall: UAS (unlabeled attachment score) og LAS (labeled attachment score).

3.2 Evaluering på andre teksttyper

Evalueringen av andre teksttyper var relativ enkelt. Før vi begynte å løse denne deloppgaven, antok vi at parseren som vi hadde trent skulle gi oss et mindre sannsynlighet, nettopp fordi filene vi brukte var på nynorsk, sammenlignet med filene som vi brukte, som var skrevet i bokmål. Her er en tabell over de forskjellige UAS og LAS resultater:

	UAS	LAS
no_bokmaal-ud-dev.conllu	0.895 ≈ 89.5%	0.805 ≈ 80.5%
no_nynorsk-ud-dev.conllu	0.695 ≈ 69.5%	0.568 ≈ 57%
no_nynorsklia-ud-dev.conllu	0.490 ≈ 49%	0.328 ≈ 33%

Figure 5: UAS og LAS

3.3 Refleksjon

Vi kan observere at vi får opptil 90 prosent dersom vi bruker bokmål filene, nettopp fordi vi trente parseren med et sett som inneholdte bokmål setninger. Vi kan se at prosenten minker og minker dersom vi bruker datasett som ikke inneholder hele bokmål setninger. Vi brukte ny-norsk istedenfor, og da minker det prosenten med ca. 20-40 prosent.