Homework 8: unification grammar

October 28, 2019

This homework gives you a taste of grammar engineering by writing linguistically-perspicuous analyses of German and Spanish. First, read chapter 9 of the NLTK book. The version on the web has been updated for python3, however it is missing some diagrams. So I have posted a scan of the old edition up on eLC.

German

Q1 Modify `german.fcfg` to incorporate the treatment of subcategorization presented in §3.1 of chapter 9. This is the the book's exercise 5. Demonstrate that your grammar gives correct results on all subcategorization classes; turn in trees e.g. by using the print to postscript menu option after issuing `draw()`

Spanish

Then consider these Spanish examples from Bresnan, Asudeh, Toivonen & Wechsler (2016).

(1) a. Juan vió algo.
saw something
"Juan saw something."

b. * Juan vió a algo.
saw Prep something
"Juan saw something."

(2) a. Juan vió a alguien.
saw Prep someone
"Juan saw someone."

b. * Juan vió alguien.
saw something
"Juan saw someone."

The symbol Prep below *a* stands for 'Preposition'. It is assumed here that *a algo* and *a alguien* in the above examples are prepositional phrases (PPs). Such an analysis is not uncontroversial, alternatively this element could be considered some form of case marker within the noun phrase.

Q2 Propose a set of phrase structure rules that will generate the grammatical sentences in 1 and 2, while ruling out the ungrammatical ones.

Although word order in Spanish is fairly free, you may assume for the purpose of this exercise that Spanish phrase structure is roughly comparable to what we find in English, and that the Spanish S and PP rules are the following

(R1) S → NP VP

(R2) PP → P NP

It is left to you to provide the VP and NP rules. Take inspiration from the English and German examples in chaper 9 of the NLTK book. The VP should include the possibility a PP daughter, and you should consider carefully the treatment of this constituent within VP. In particular, bear in mind the feature specifications on the nouns e.g. L2 and L3 below.

(L1) `P[case=acc, +anim] -> ’a’`
(L2) `N[-anim] -> ’algo’`
(L3) `N[+anim] -> ’alguien’`
(L4) `N[num=sg,gen=masc,+anim] -> ’Juan’`
(L5) `V[tense=past] -> ’vió’`

Test your modified grammar using the testing utility provided in `hw8-spanish.py` shown below

```
from nltk import load_parser

# HW8 Q2
onea = ’Juan vió algo’ # ok
oneb = ’Juan vió a algo’   # star
twoa = ’Juan vió alguien’  # star
twob = ’Juan vió a alguien’ # ok

testsuite = [onea,oneb,twoa,twob]

# set cache to False to force it to reload modified grammar
p = load_parser(’spanish-modify-this-grammar.fcfg’,cache=False)

# utility that prints out the number of analyses
# that the parser finds
# for each member of the test suite defined above
def check(p):
    print(’{:<20}\t{:>8}’.format(’example’,’analyses’))
    f = lambda s: len(list(p.parse(s.split())))
    for ex in testsuite:
        print(’{:<20}\t{:>8}’.format(ex,f(ex)))
```

Turn in your grammar, along with an cogent paragraph explaining your analysis.