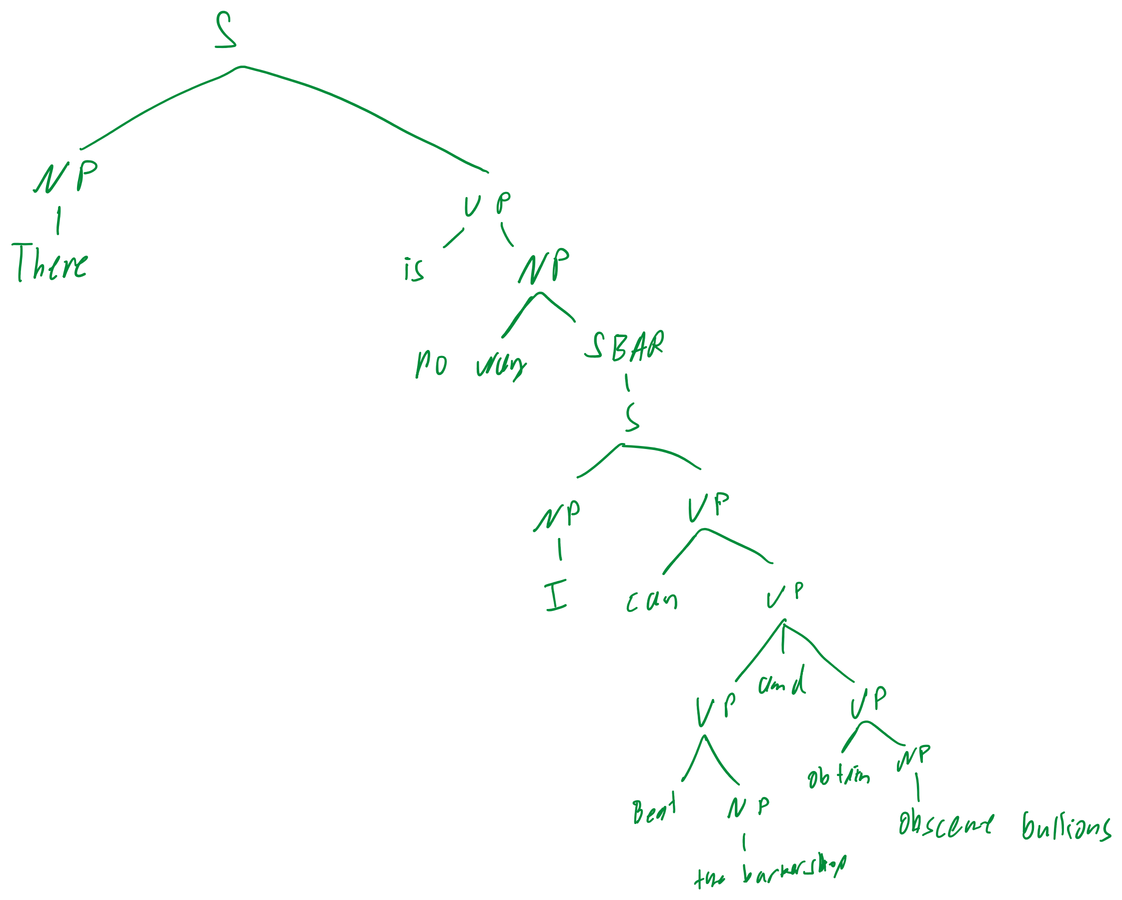
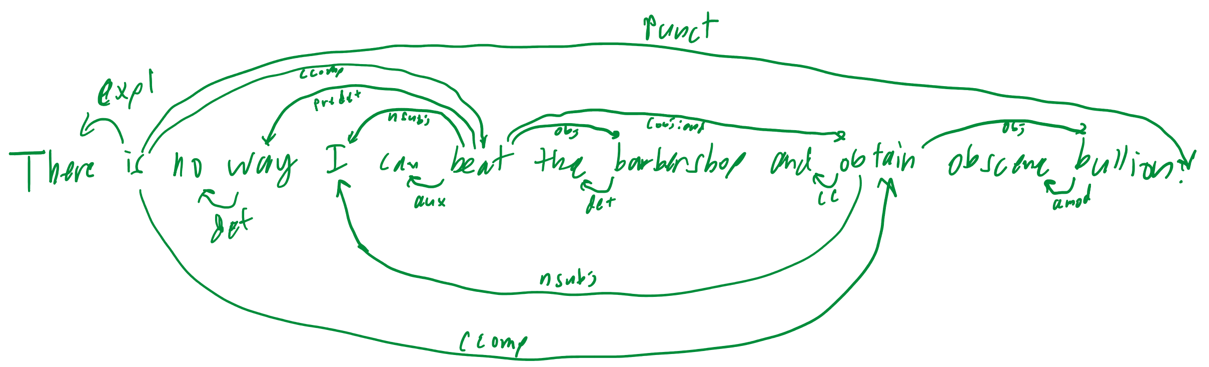
Sentence Parsing

Sunday, October 16, 2022 2:11 PM



NP: Noun Phrase
A phrase that functions as a noun
VP: Verb Phrase
A phrase that functions as a verb
SBAR: a clause introduced by a possible empty subordinating conjunction

S: a simple declarative clause that does not exhibit subject-verb inversion, excluding SBAR's or SBARQ's


punct
expl
ccomp
predet
nsubj
obj
conj
obj
There is no way I can beat the barbershop and obtain obscene bullions!
det
aux
det
cc
amod
nsubj
ccomp

Expl: Expletive

Existential there

Det: determiner

Relation between head of NP and its determiner

Aux: Auxillary

The non-main verb of a clause

Cc: coordination

Relation between conjunct element and the coordination conjunction word of the conjunct

Amod: adjectival modifier

Any ajdectival phrase that serves to modify the meaning of the NP

Nsubj: nominal subject

A noun phrase that is the syntactic subject of a clause

DObj: direct Object

The noun phrase as part of a VP which is the accusative object of the verb

Conj:and: Conjunct

What the “and” is conjoining

predet: predeterminer

Relation between the head of an N and a word that precedes and modifies the meaning of the NP Determiner

Ccomp: Clausal Complement

Dependent clause with an internal subject which functions like an object of the verb or adjective

Punct: punctuation

It’s, a punctuation…?!

```
{'verb': 'is', 'description': 'There [V: is] [ARG1: no way I can beat the barbershop and obtain obscene bullion] .', 'tags': ['O', 'B-V', 'B-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'I-ARG1', 'O']}
{'verb': 'can', 'description': 'There is no way I [V: can] beat the barbershop and obtain obscene bullion .', 'tags': ['O', 'O', 'O', 'O', 'O', 'B-V', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']}
{'verb': 'beat', 'description': 'There is [ARGM-MNR: no way] [ARG0: I] [ARGM-MOD: can] [V: beat] [ARG1: the barbershop] and obtain obscene bullion .', 'tags': ['O', 'O', 'B-ARGM-MNR', 'I-ARGM-MNR', 'B-ARG0', 'B-ARGM-MOD', 'B-V', 'B-ARG1', 'I-ARG1', 'O', 'O', 'O', 'O', 'O']}
{'verb': 'obtain', 'description': 'There is [ARGM-MNR: no way] [ARG0: I] [ARGM-MOD: can] beat the barbershop and [V: obtain] [ARG1: obscene bullion] .', 'tags': ['O', 'O', 'B-ARGM-MNR', 'I-ARGM-MNR', 'B-ARG0', 'B-ARGM-MOD', 'O', 'O', 'O', 'O', 'B-V', 'B-ARG1', 'I-ARG1', 'O']}
```

There [V: is] [ARG1: no way I can beat the barbershop and obtain obscene bullion] .

There is no way I [V: can] beat the barbershop and obtain obscene bullion .

There is [ARGM-MNR: no way] [ARG0: I] [ARGM-MOD: can] [V: beat] [ARG1: the barbershop] and obtain obscene bullion .

There is [ARGM-MNR: no way] [ARG0: I] [ARGM-MOD: can] beat the barbershop and [V: obtain] [ARG1: obscene bullion] .

ARG1 is the thing that is being acted upon which is "The Barbershop" and "Obscene Bullion"
ARG0 is the actor upon the thing, which is "I"
ARGM-MNR pertains to the how which is "way" or rather "no way"

In terms of parsing methods I can only really recommend/prefer the PSG tree as it displayed the information in the more readable format compared to the others, I dislike the incredibly obfuscated method that SRL uses as it requires you to learn arbitrary terms that don't make a whole lot of sense, while the other methods at least use abbreviations that could relatively reasonably be extrapolated from.