

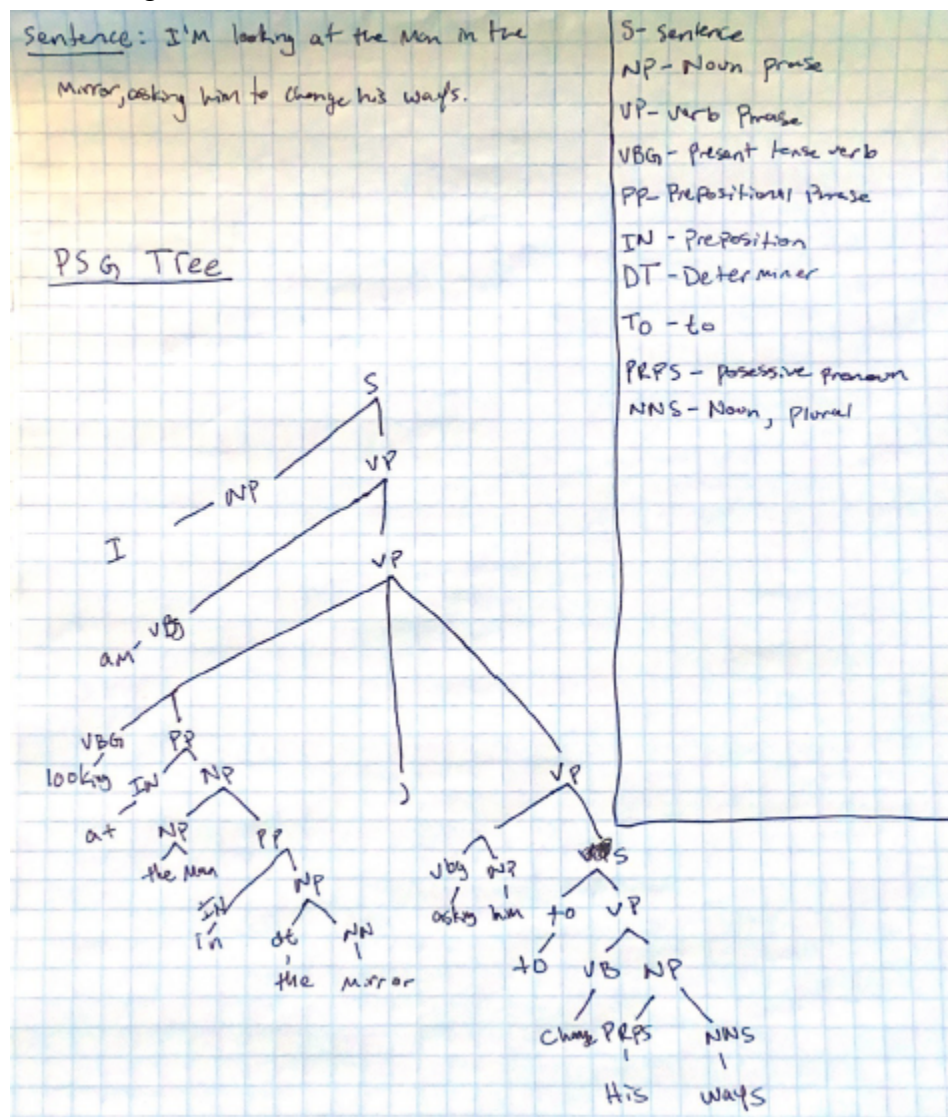
Sentence Parsing

1. Introduction:

In this document I will explore testing various sentence parsing methods to interpret the sentence
"I'm looking at the man in the mirror, asking him to make a change"
to see how each different parsing method breaks down the sentence.

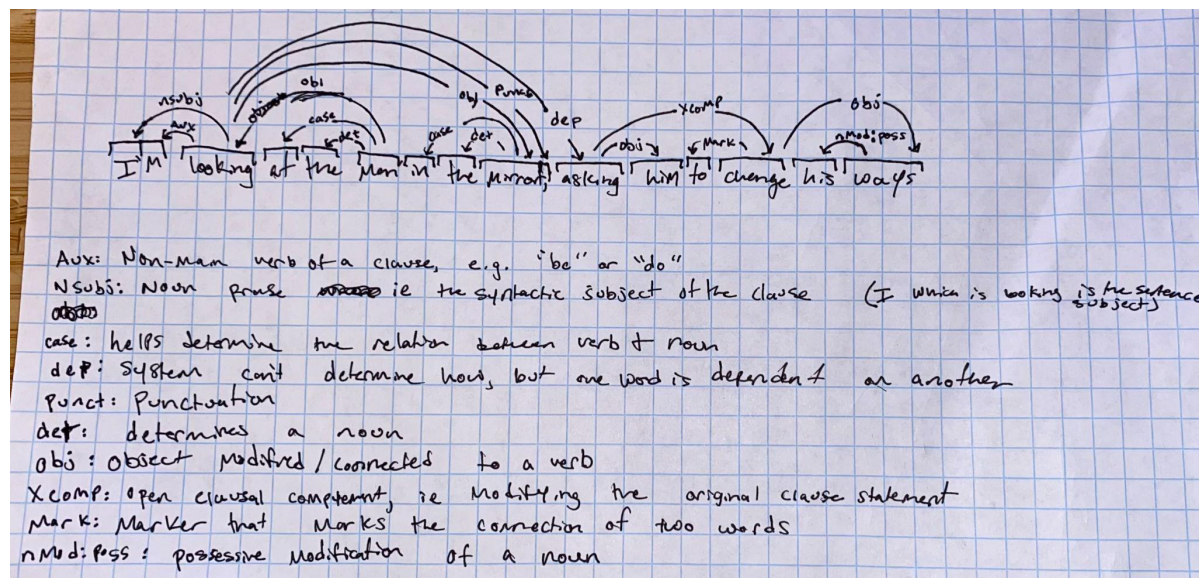
2. PSG Parsing:

Below is a PSG parsed tree created from the above sentence. On the right is the descriptors for the POS tags.



3. Dependency Parsing:

Below is a hand drawn dependency breakdown of the sentence.



4. SRL Parsing:

Interpreting this manually, we would get the following outputs:

```
[ARG0: I] 'm [V: looking] [ARG1: at the man in the mirror]
, [ARGM-PRD: asking him to change his ways] .
```

Predicate: looking

ARG0: I : I am the agent of this sentence, doing the action of looking

ARG1: at the man in the mirror : Passive actor; the man in the mirror

ARGM-PRD: asking him to change his ways : doing the asking

Modifiers:

PRD: an adjunct of a predicate that is in itself capable of carrying some structure

```
[ARG0: I] 'm looking at the man in the mirror , [V: asking]
[ARG2: him] [ARG1: to change his ways] .
```

Predicate: asking

ARG0: I : I am the agent of this sentence, doing the action of asking

ARG1: to change his ways : the subject of change is argument 1

ARG2: him : instrument of ask, the man being asked

I 'm looking at the man in the mirror , asking [ARG0: him]
to [V: change] [ARG1: his ways] .

Predicate: change

ARG0: him : he is the agent of this sentence, he is the one changing

ARG1: his ways : his ways are being affected , changed by the verb

5. Comparing the parsing methods:

After working with all three of these algorithms, two stood out to me the most. While PSG parsing is nice to visualize the breakdown of a sentence structure, it seems more like a foundational function than something to be interpreted. What I mean by that is that the other two algorithms (Dependency and SRL) produce an output that is easier to use in creating a model. With dependency parsing, we see which words correlate, which allows us to more easily “understand” an inputted sentence. Similarly, SRL is like a shallow semantic parser that outputs the subject, verb, and possible rules for a sentence. I believe this would be the most useful of the 3 algorithms to use for our chatbot project, since it allows us to create a ruleset based on the corpora and sentence database we have created.