

INTJ: An interjection, corresponds to UH UH: Interjection (word level)

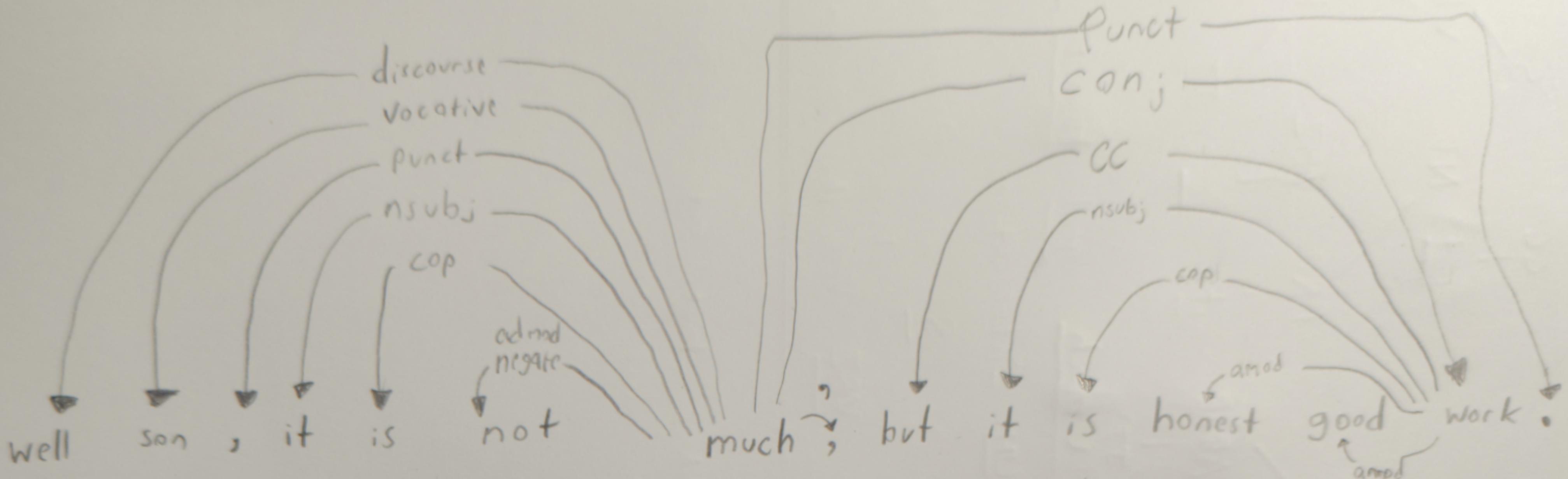
NP: Noun Phrase NN: Noun (singular or mass) S: Simple Declarative

PRP: Personal pronoun VP: Verb Phrase VBZ: Verb, third person singular
Present clause

JJ: Adjective CC: Coordinating Conjunction

Note: I counted the punctuation as tokens, but out of fear of
not being complex I will add the adj "good"

RB: Adverb ADJP: Adjective phrase



discourse(well, much): Used for interjections, in this case marking the start of discourse.

vocative(son, much): Mark a dialogue participant being addressed

punct(, , much): Just to retain punctuation,(punctuation).

nsubj(it, much): Syntactic subject of a clause (The root of the clause is a complement of a copular verb, so the governor is an adj)

cop(much, is): Marks (the copula is the dependent of its complement) the copula complement of a copula verb

negate(much, not): Relation between the negation word and what it negates

negate(much, not): Relation between two elements that are connected by a coordinating conjunction

conj(much, work): Relation between an element of a conjunct and a coordinating conjunction
(work) (but)

The root is work because but comes at the start of the sentence

amod(honest, work): Modifies the meaning of the noun phrase

amod(good, work):

Well Son, it is not much, but it is honest good work,

Frames: We have the copular verb 'is' in two frames,
so we have two predicates, both are 'is'

1: it is not much
arg-1 arg-neg arg-2

arg1 - it it is the agent, doing the action (of being not much)
arg-neg-not not modifying much to be negative
arg2 - much much is an attribute of the subject

2: but it is honest good work

arg¹ - it it is the agent
arg² - honest good work is the attribute of the subject

I realize I should have picked a sentence with more action...
but it's honest good work...

Of the algorithms used in this exercise, I will say that the SRL parser was the easiest for me to interpret only because of its simplicity. It requires a lot of annotation of syntactic rules so it must struggle with more figurative outside the box sentences. Yet, its result is a labeling system that can annotate the complex embedded syntax we often use. On the other hand, PSG's are able to handle interpreting the entire structure of data, getting better the more its model is fed data. That can lead to accurate results that take into account the entire structure of a language. Yet it also means that complex sentences, with more and more embedded clauses might not get identified on the edge of the model's capabilities. The dependency parse method definitely wasn't as easy to interpret as the others, yet it did seem the best at granularly understanding every part of a sentence. It was able to accurately explain the interjection/vocative relationship to the predicate, and that dependency was helpful. Yet, I think for natural language purposes, I'm more interested in grammar and info extraction so I liked the simplicity of SRL.