MXS170018

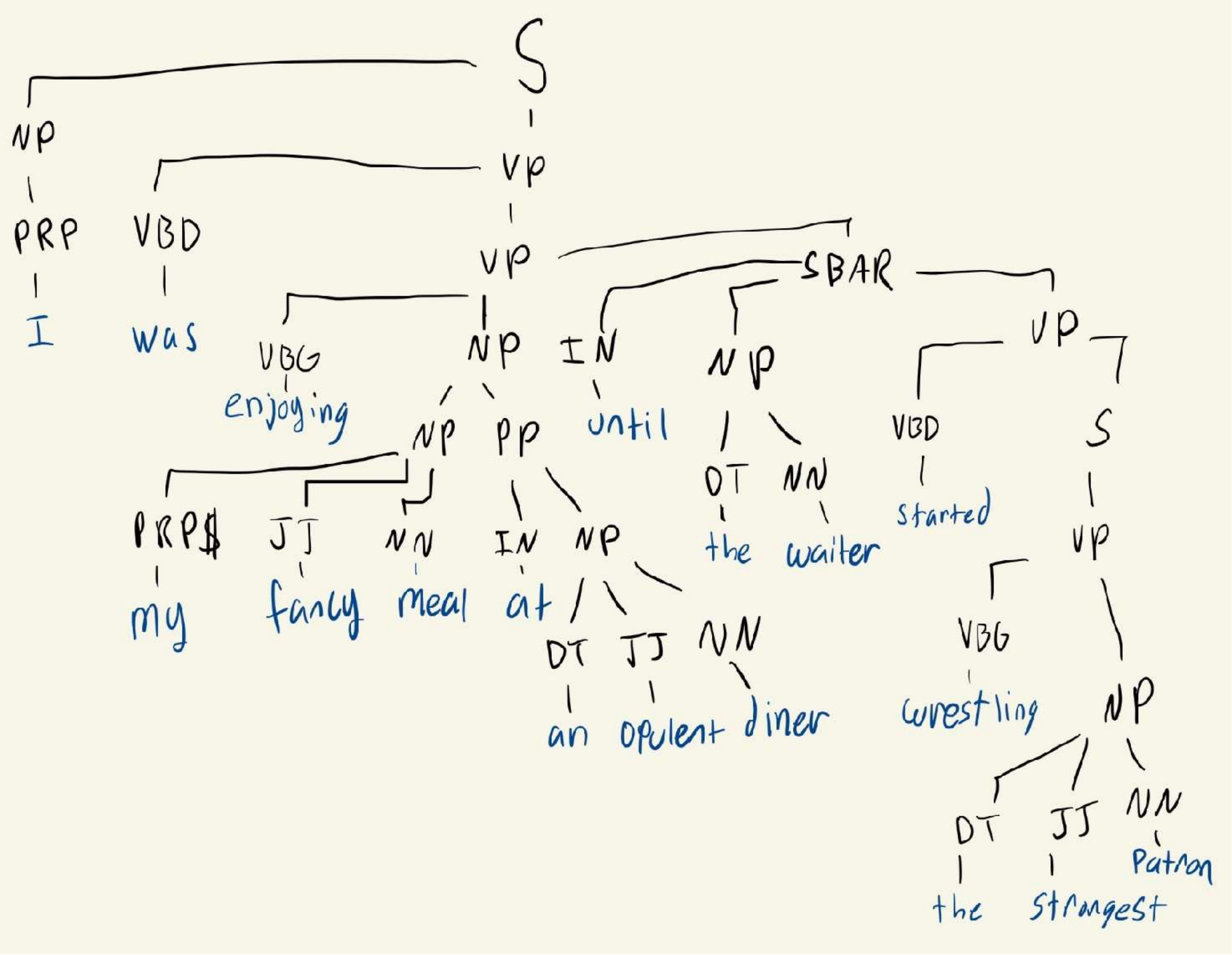
Manuel Salado Alvarado

4395.001

Sentence Parsing Comparison

I created a sentence and drew the Constituency Parse Tree, the Dependency Parse Graph, and the Semantic Role Label Parse Arguments. “I was enjoying my fancy meal at an opulent diner until the waiter started fighting the strongest patron” was the sentence chosen because it has 18 tokens and more than one clause.

Constituency Parse



S – Simple declarative clause, essentially the entire sentence

SBAR – Clause introduced by a subordinating conjunction: ,So ; ,But; ,Until

NP – Noun Phrase, a phrase that contains a noun

VP – Verb Phrase, a phrase that contains a verb

PRP - Person Pronoun: I, Me

VBD – Verb, Past Tense: -ed

VBG – Verb, Gerund or present participle: -ing

PRP$ - Possessive Pronoun: my

JJ – Adjective

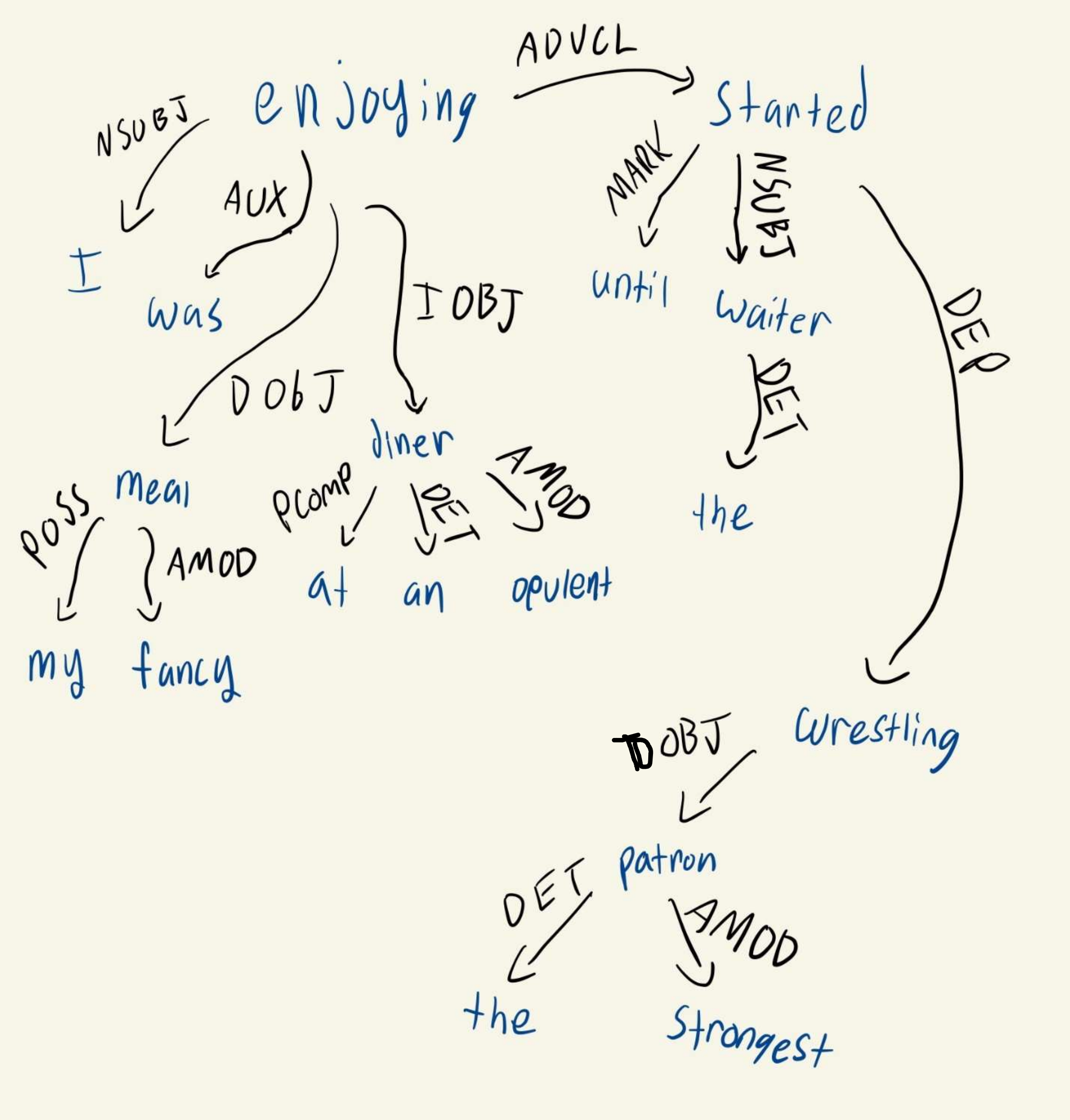
NN – Noun, singular or mass

PP – Prepositional Phrase – Phrase with a preposition

IN – Preposition or Subordinating Conjunction: Over, In or So, But

DT – Determinate: The

Dependency Parse



ADVCL – Adverbial Clause modifier: clause modifying the verb

AMOD – Adjectival Modifier: Adjective that modifies the noun in the noun phrase

AUX – Auxiliary: non-main verb of the clause: To have/be/do

DEP – Dependent, mostly used when another dependency can’t be identified: Started Wrestling

DET – Determiner, the relation between the head of a noun phrase and its determiner

DOBJ – Direct Object, the direct object of a verb phrase is the noun phrase

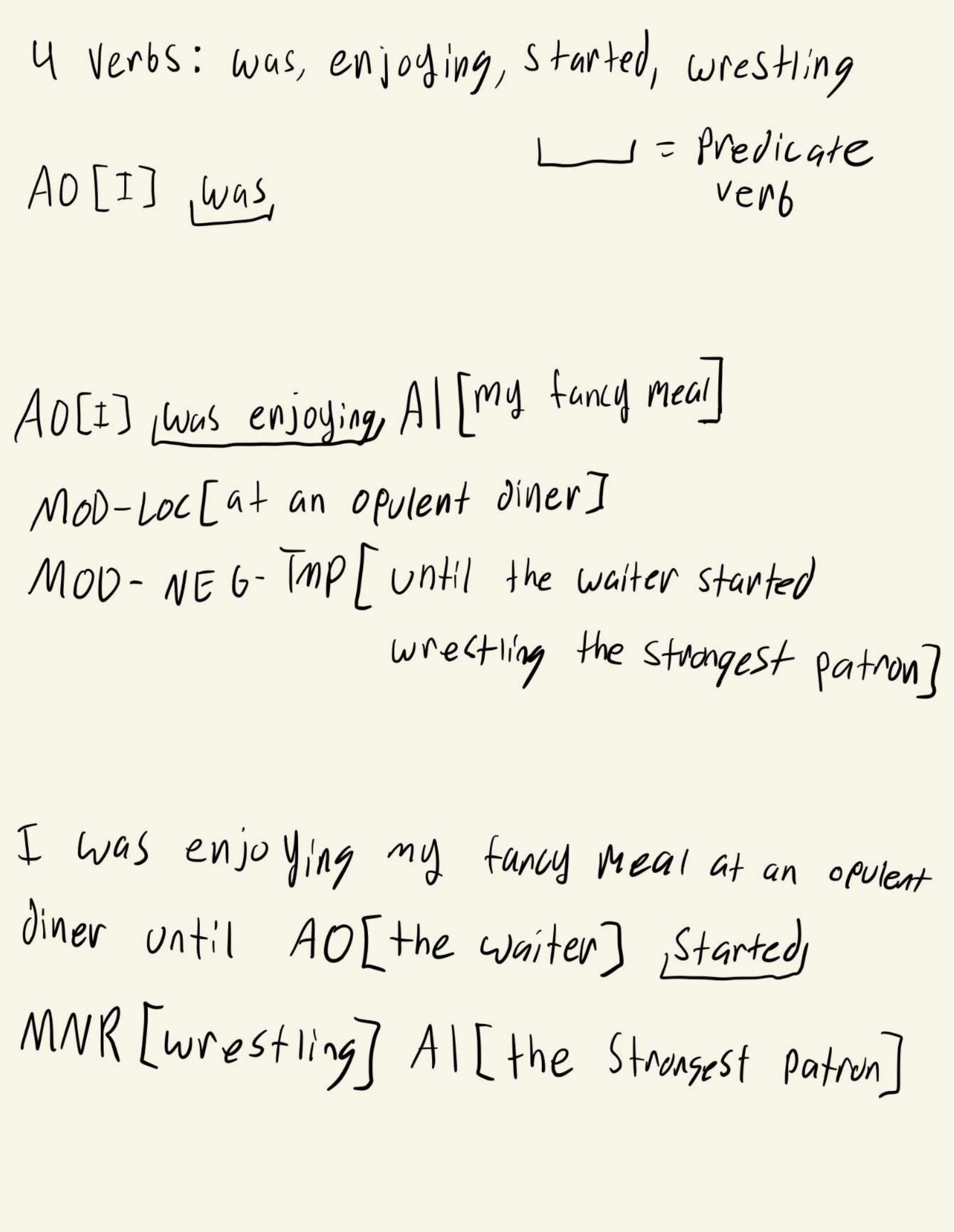
IOBJ – Indirect Object, the indirect object of a verb phrase is the noun phrase

NSUBJ – Nominal Subject, noun phrase which is the syntactic subject of a clause.

PCOMP – Prepositional Complement, the complement of a preposition

POSS – Possession Modifier: their, his, hers

Semantic Role Label Parse



4 Verbs: was, enjoying, started, wrestling

└──┘ = Predicate verb

A0[I] was

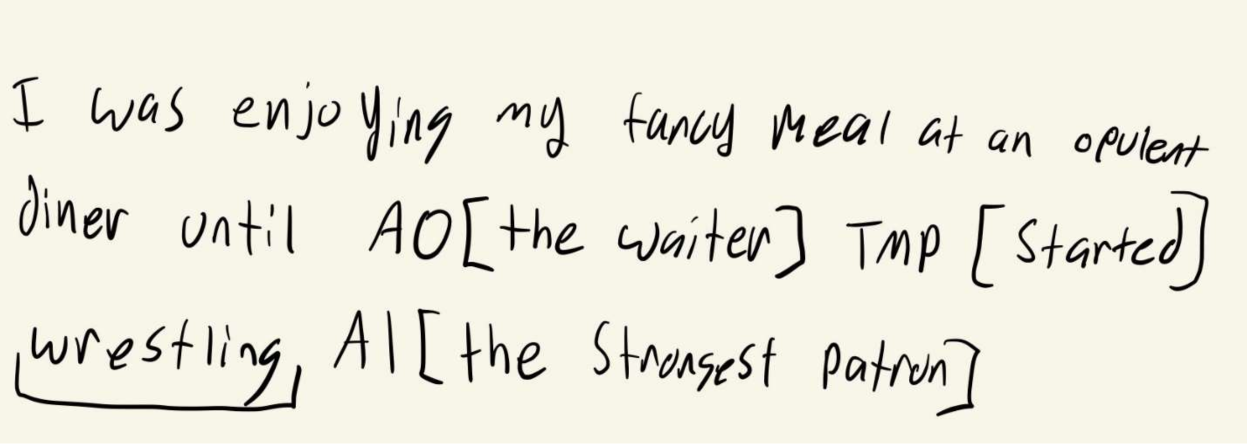
A0[I] was enjoying A1[my fancy meal]

MOD-LOC[at an opulent diner]

MOD- NEG- TMP[until the waiter started wrestling the strongest patron]

I was enjoying my fancy meal at an opulent diner until A0[the waiter] Started

MNR[wrestling] A1[the Strongest patron]



I was enjoying my fancy meal at an opulent diner until A0[the waiter] TMP [Started] wrestling, A1[the Strongest patron]

WAS:

A0: I ; main agent for WAS

ENJOYING:

A0: I ; main agent for enjoying

A1: my fancy meal; the passive actor in enjoying my fancy meal

LOC: at an opulent diner: where the enjoying happened

NEG-TMP: until the waiter started wrestling the strongest patron; when the enjoying stopped

STARTED:

A0: the waiter; main agent for started

A1: the strongest patron; the passive actor in the waiter started wrestling the strongest patron

MNR: wrestling; how the starting was performed

WRESTLING:

A0: the waiter, main agent for wrestling

A1: The strongest patron; the passive actor in the waiter started wrestling the strongest patron

TMP: started; when the wrestling happened

Sentence Parsing Method Comparison

The Constituency Parsing Tree was the easiest to discern because of the widest availability of tags. It does an excellent job at identifying the different components of the sentence. Unfortunately, it isn’t too terribly useful by itself. If I were to replace every word within my sentence with a different word that matches the POS tag, it would be mostly gibberish. There needs to be a stricter dependency between the tags and the rest of the sentence. The Dependency Parse Graph works well at bridging the gap. It makes it so that every word is dependent on another word in the sentence. Constituency parsing almost does this by making a tree, but there isn’t a strict dependency on the parent of the leaf. This helps place the verbs at the forefront to analyze its dependencies. If isolating verb dependencies is your goal, then Semantic Role Parse will also be useful. It identifies the verbs and the arguments that make up the verbs as well as the modifications on the verb. It is unfortunate that such a powerful tool is limited by shaky definitions and 7 arguments seems a tad small for more complex sentences. That is why it is useful to separate more complex sentences into smaller verb fragments. The modifiers and arguments could still be fleshed out more. Generally, I believe that a Constituency Parsing Tree with a Dependency Parse Graph would be an excellent way of combing two parses to potentially make a conversation more realistic by being able to properly listen and understand and being able to make a sentence.