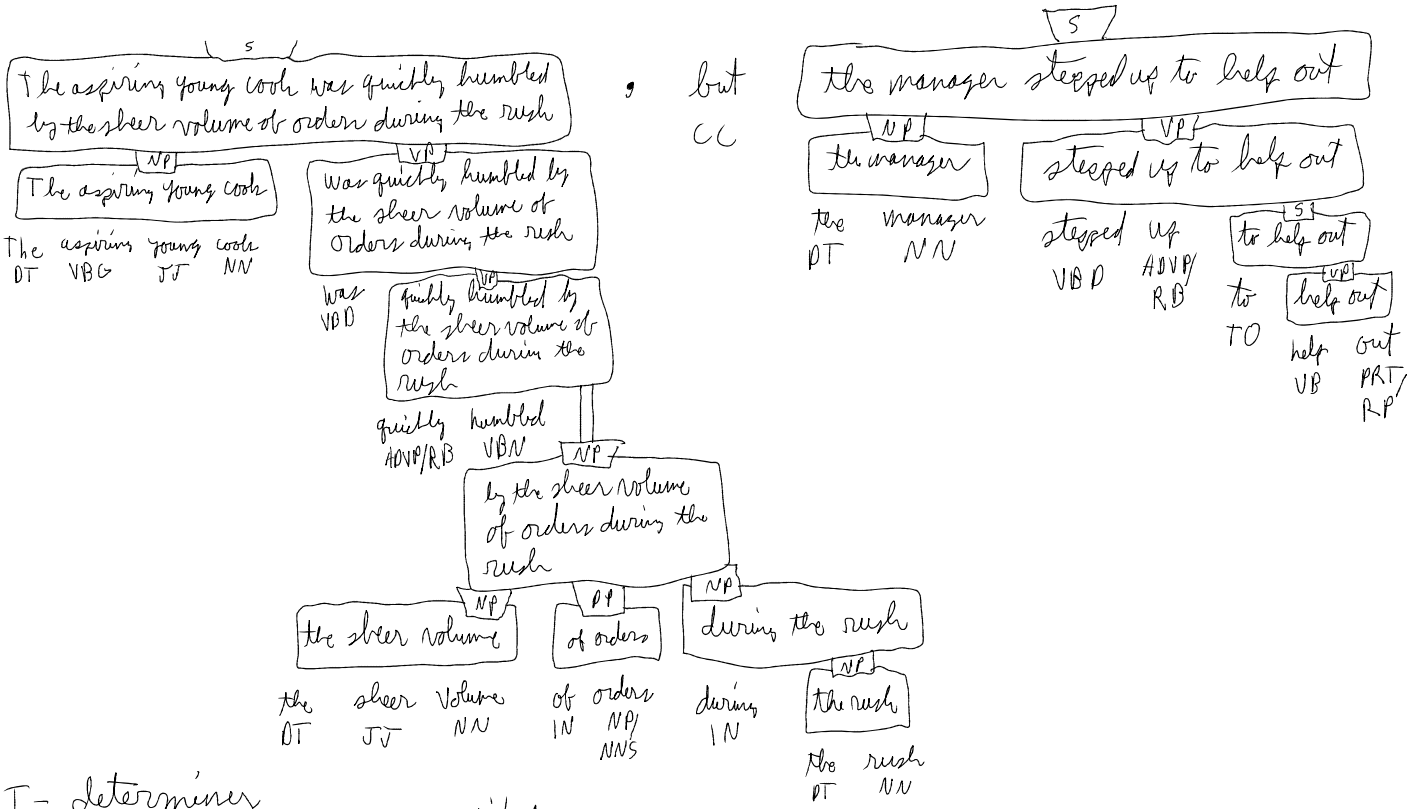
Sentence Parsing
Friday, March 3, 2023 4:05 PM

The aspiring young cook was quickly humbled by the sheer volume of orders during the rush, but the manager stepped up to help out.

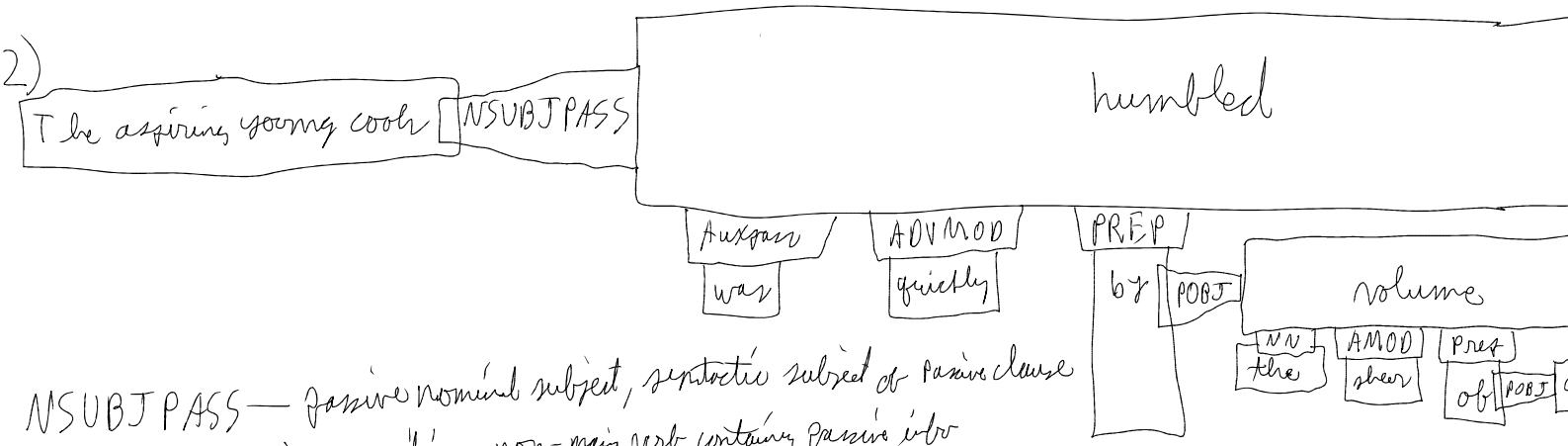
1)



DT - determiner
VBG - Verb, gerund, or present participle
JJ - adjective
NN - Noun, singular or mass
VBD - verb, past tense
ADVP - adverb phrase
RB - adverb
VBN - Verb, past participle
IN - Preposition or subordinating conjunction
NNS - noun, plural
S - simple declarative clause
NP - Noun phrase
VP - Verb phrase
PP - prepositional phrase
CC - Coordinating conjunction
TO - the word "to"
VB - verb, base form
PRT - Particle; category for words tagged "RP"

RP - Particle

2)



NSUBJPASS — passive nominal subject, syntactic subject of passive clause
AUXPASS - passive auxiliary, non-main verb contains passive info
ADVMOD — adverb modifier, modifies meaning of adverb
PREP — prepositional modifier, serves to modify meaning of other POS
POBJ — object of a preposition, head of noun phrase following preposition
NN — noun compound modifier, noun that modifies head of noun
AMOD - adjectival modifier, adjective phrase modifies meaning of NP
DEP - dependent, default case if unable to determine relation
DET - determiner, relation between head of NP & determiner
NSUBJ - nominal subject, syntactic subject of a clause
XCOMP - open clausal component, predicative or clausal complement without its own subject

3)

a) aspiring

The | aspiring (V) | young cook (ARG0) | was quickly

humbled by the sheer volume of orders during the rush, but the manager stepped up to help out.

a) The young cook is the agent who is aspiring

b) was

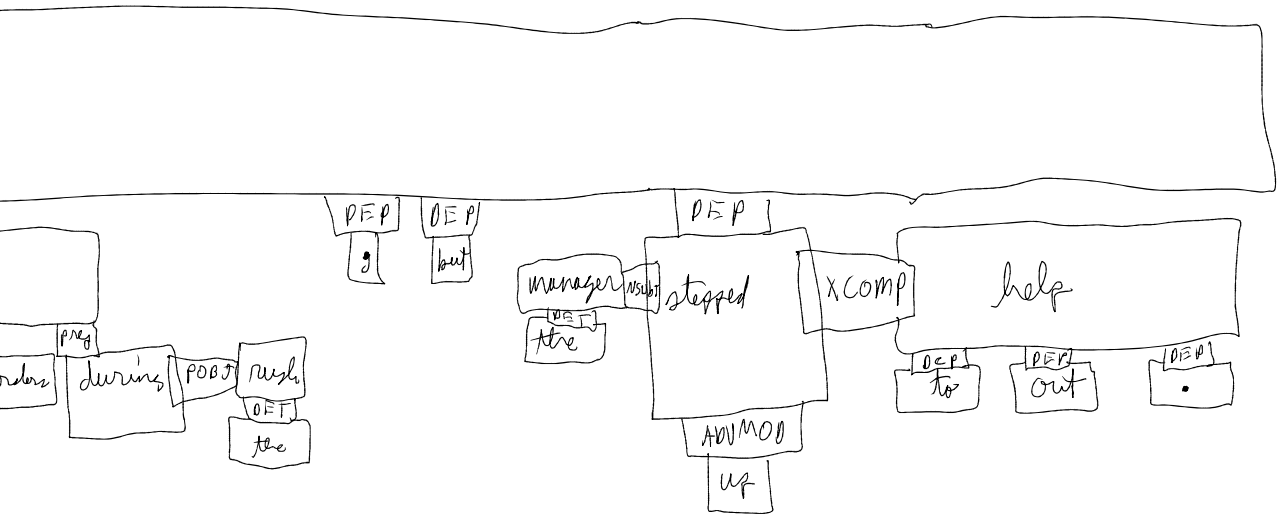
The aspiring young cook | was (V) | quickly humbled

by the sheer volume of orders during the rush, but the manager stepped up to help out.

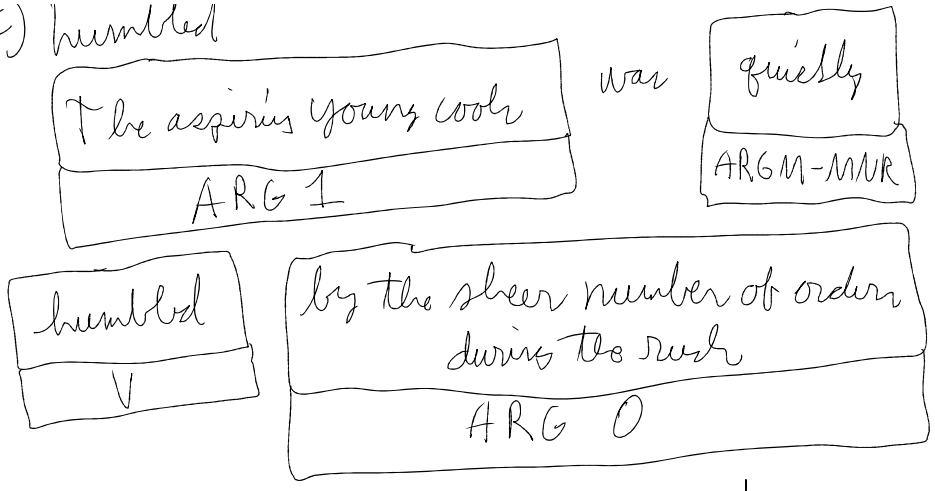
b) due to "was" being merely a linking verb, its presence is merely acknowledged by the model

c) humbled

c) the orders and associate


DEP
g
DEP
but
DEP
manager
NSUBJ
stepped
XCOMP
help
DET
the
PREP
during
POBJ
rush
DET
the
ADVMOD
up
DEP
to
DEP
out
DEP
.

c) humbled

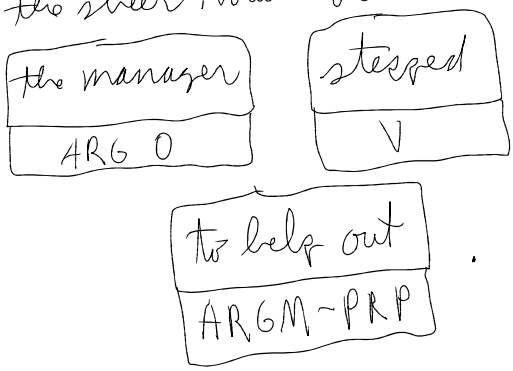


, but the manager stepped up to help out.

c) the orders and associate words were doing the humbling, quickly modifies how the humbl was being done, and it was performed on the aspiring young cook.

d) stepped

The aspiring young cook was quickly humbled by the sheer volume of orders during the rush, but



.

d) The manager was the agent performing the stepping (up), and the prepositional phrase "to help out" assigns meaning to how the manager stepped.

e) help

The aspiring young cook was quickly humbled by the sheer volume of orders during the rush, but

e) the manager was the agent doing the helping. I imagine if "to help out contained "him", then the aspiring young cook would be identified as ARG 1.

ARG 0 – proto-agent, causer of an event

ARG 1 – proto-patient, undergoes change/causally affect

ARGM-MNR — how the action was performed

ARGM-PRP — Reason for action

For the chosen sentence "The aspiring young cook was quickly humbled by the sheer volume of orders during the rush, but the manager stepped up to help out.", I believe that the dependency parse captures the best representation of the overall meaning of the sentence. Relative to the constituency parsing/PSG, while the grammatical information seems less well represented, it still seems to be captured sufficiently while having the benefit of, in my opinion, a better flow to represent the construction and relations between words in the sentence- the nature of the PSG tree having one or a few words split off at each level makes it feel clunky and inefficient overall. The semantic role labeling seemed adequate for its purpose of gleaning more surface-level information from the sentence. While I can imagine it being incredibly useful in situations where you have a very large corpus and can only store so much semantic information about each sentence, in this case it felt like it left too much on the table in this case, so to speak. I feel that if I were to query information about the sentence in a context where the model has a representation in SRL vs a Dependency Parse, I would quickly find the limit of what it could answer with the SRL whereas with the dependency I would more likely struggle to find a question it could not answer.