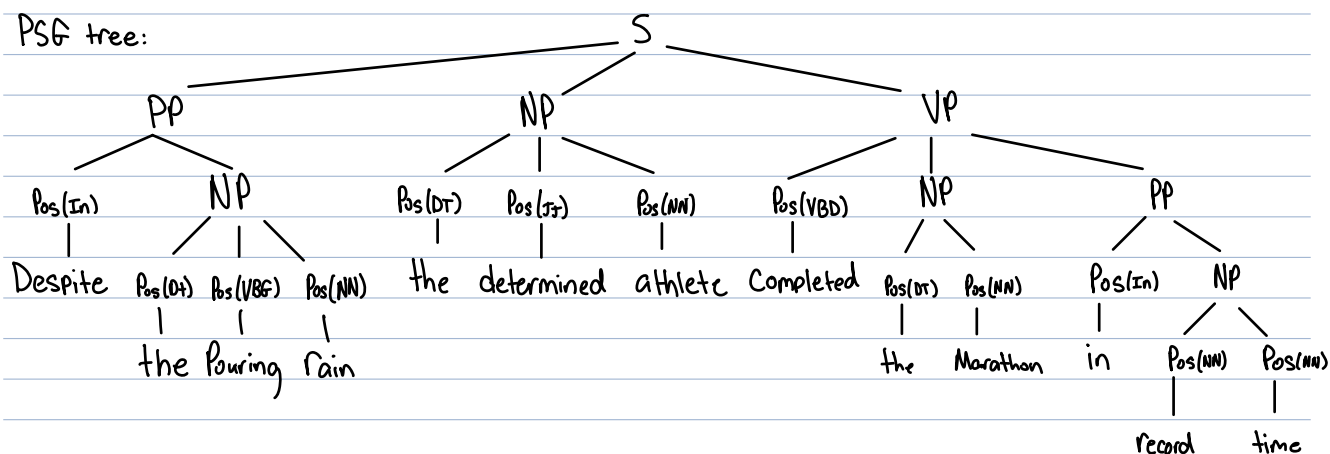


Sentence: Despite the pouring rain, the determined athlete completed the marathon in record time.

PSG tree:



PP: Despite the pouring rain

NP: the pouring rain

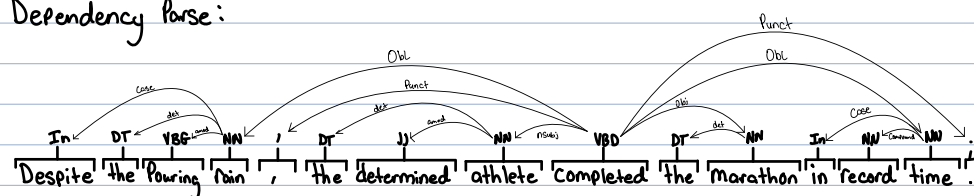
NP: the determined athlete

VP: Completed the marathon in record time

NP: the marathon PP: in record time

NP: record time

Dependency parse:



det: determiner - relation between the head of NP and its determiner

amod: adjective modifier - any adjective phrase that serves to modify the meaning of the NP.

Punct: punctuation - any piece of punctuation in a clause

Nsubj: nominal subject - noun phrase which is the syntactic subject of a clause.

Obj: direct object - object of VP is noun phrase which is the accusative object of the verb.

Compound: noun compound modifier - any noun that serves to modify head noun.

## SRL Parsing:

Predicate = Pouring (Verb)

Argument 0: rain → Since the rain is Pouring

Predicate = Completed (Verb)

ARGM-ADV: Despite the Pouring rain → Adverbial modifier Use to modify Part of the Sentence

Argument 0: the determined athlete → the determined athlete is who did the action

Argument 1: the marathon → the marathon was what the action related to.

Argm-temp: in record time → temporal expression tells us when something happened

After using the constituency, dependency, and SLR parsers on my selected sentence they all had their own pros and cons. Starting with the constituency parse which was used to make the PSG tree. The parser was fairly straightforward to understand and identify the POS and phrase terms. Overall the parse was fair positive the only downside could be when there are ambiguity in the sentence which could cause errors in the PSG trees. As for the dependency parse I found it a bit harder to initially understand and identify the actual dependency. Also the description of the individual types of dependency relationships was hard to understand without the Stanford Dependency Manual. However, the parser did do a good job at identifying the dependency throughout the sentence even with more than one clause. Finally for the SRL parser I feel that it was under utilized for my particular sentence and I feel that a different sentence could have showcases its use better. Overall when I used the SRL parsing it was straightforward and it did a great job at identifying the predicates and their associates modifiers. However I felt that the the parse was very detailed since some of the modifier and arguments where whole phrases instead of evaluating each individual token in the sentence.