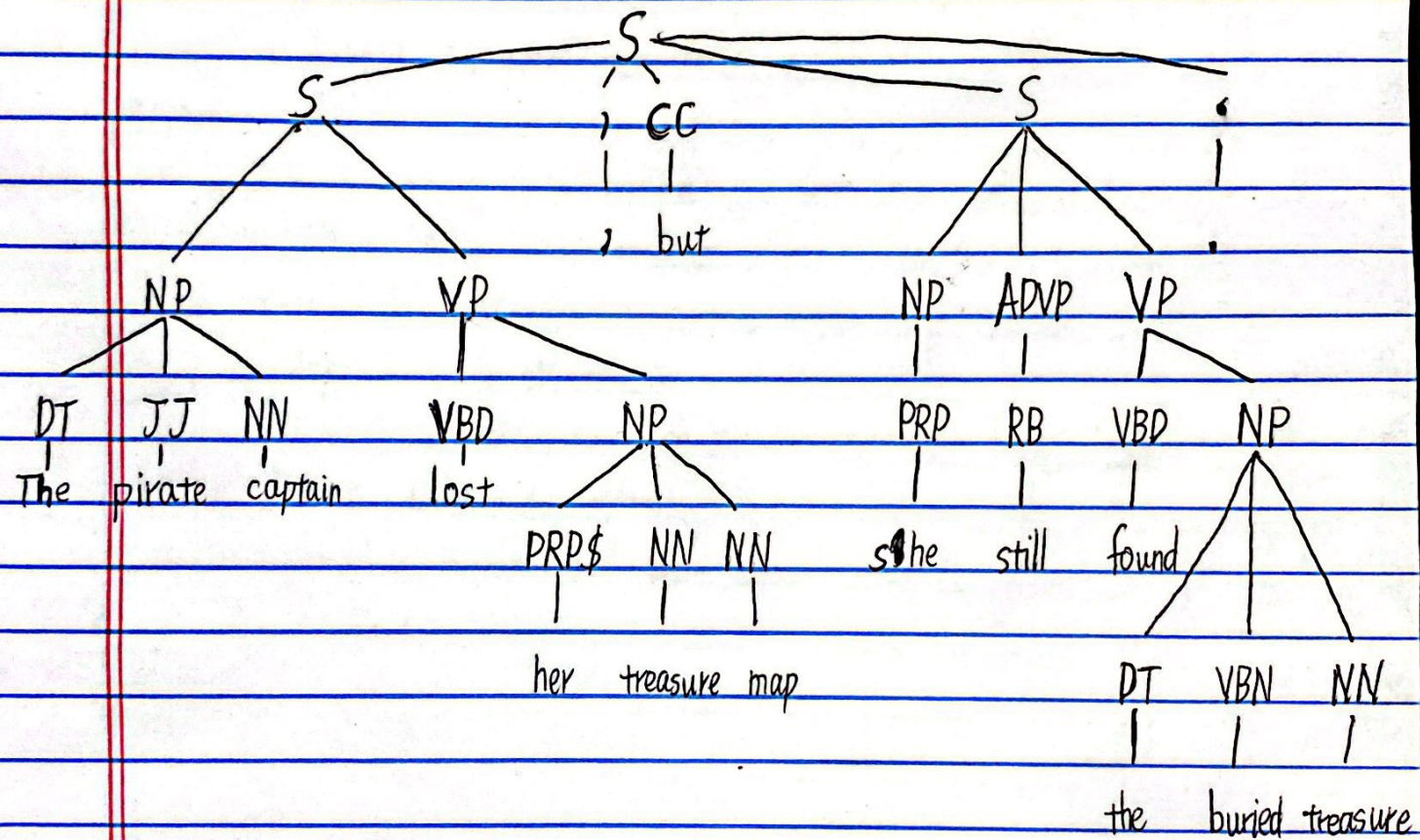
qtg190000

Sentence:
The pirate captain lost her treasure map, but she still found the buried treasure.

PSG Tree:

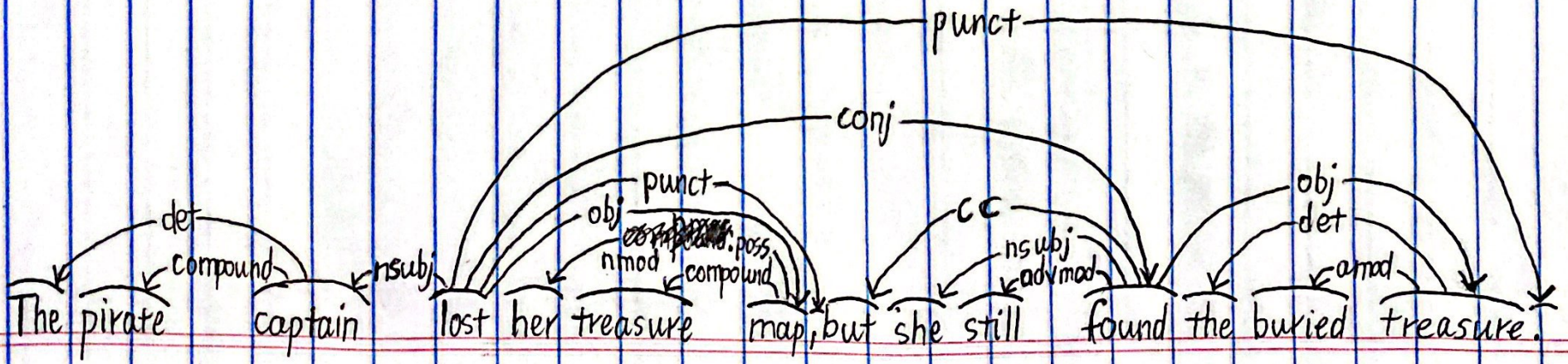



~~Phrase Keys~~ Phrase Terms:
S: Declarative Clause
~~CC: Coordinating Conjunction~~
NP: Noun Phrase
VP: Verb Phrase
ADVP: Adverb Phrase

Dependency terms:

advmod: adverb modifier
amod: adjectival modifier
cc: coordination
conj: conjunction
det: determiner
compound: noun of 2 or more words
nsubj: nominal subject
nmod: nominal modifier
obj: object
poss: possession modifier
punct: punctuation

Dependency Parse:



SRL Parse

Predicates: lost, found, buried.

1. The pirate captain lost her treasure map
   ARG0 V ARG1

ARG 0 is the agent doing the action, ARG 1 is the patient of the action.

2. she still found the buried treasure
   ARG0 ARGM-TMP V ARG1

ARG0 is the agent doing the action, ARG 1 is the patient of the action

3. buried treasure
   V ARG1

ARG1 is the patient of the action

SRL modifiers:

DIR: Direction

LOC: Location

MNR: How the action was performed

TMP: When the action happened

CAU: Reason for action

PNC: Motivation for action

Pros and Cons

The PSG Tree is easy to read in its tree format and its labels make it easy to comprehend a sentence's structure. However, it is only good for simple sentences because the tree would get too big in terms of depth and branch when we input a longer and more complex sentence.

The dependency parse excel at demonstrating the relations between the verbs and its surrounding tokens. However, it is difficult to read for long and complex sentences due to the number of relations overlapping each other.

The SRL parse is able to clearly pick out frames based on each predicate to form independent clauses. However, not all frames can be formed into an independent clause due to it not being able to differentiate adjective from verb.