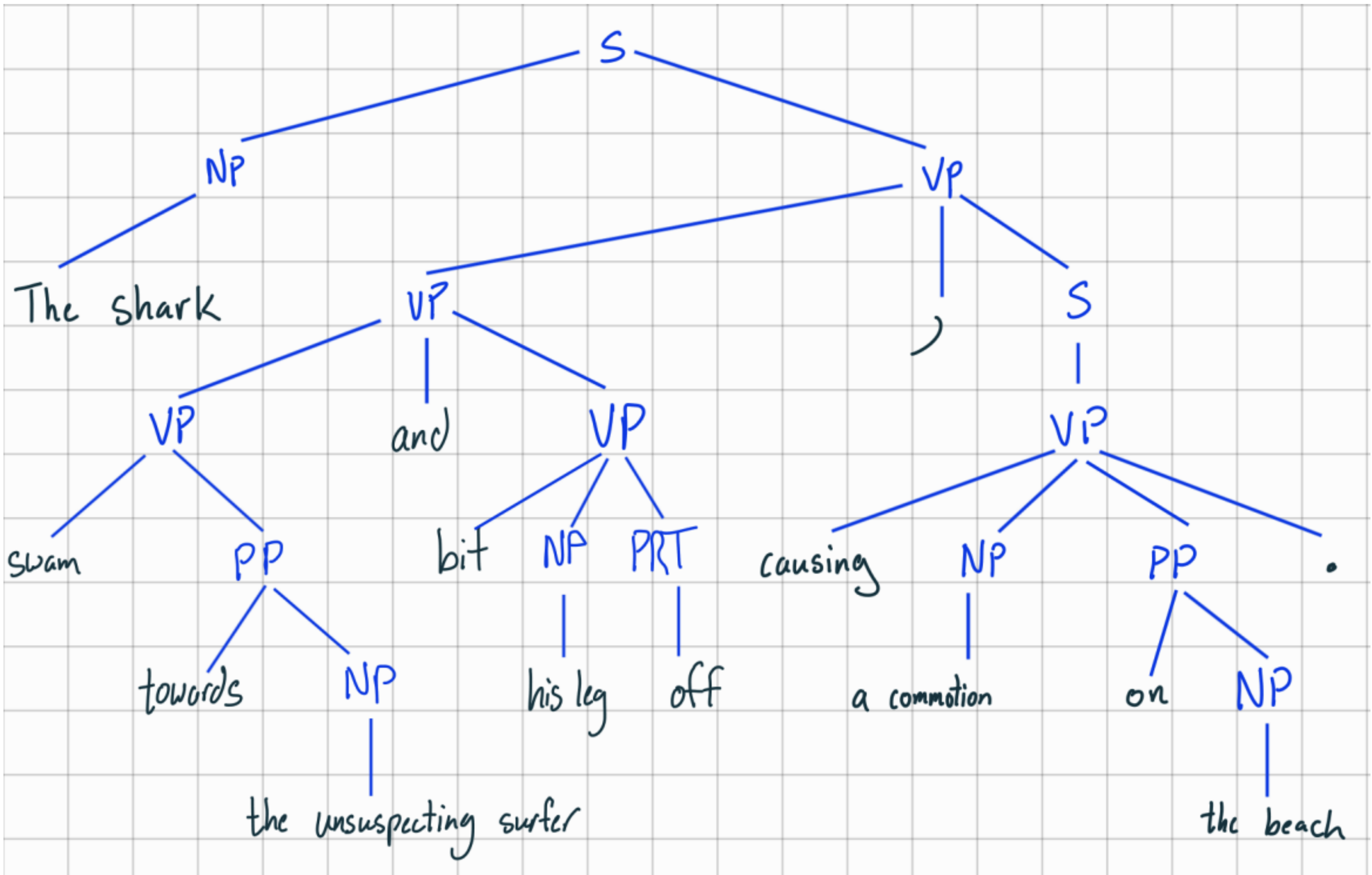
Abdullah Hasani
AHH190004

Sentence Parsing

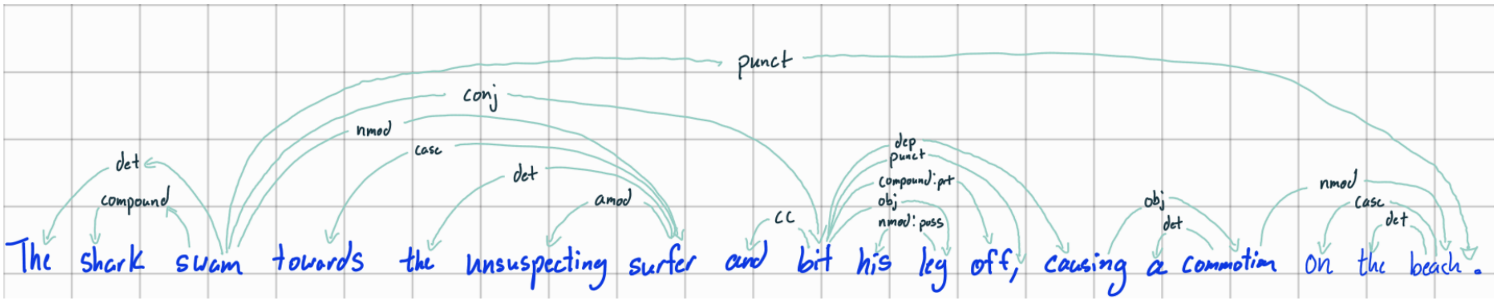
1. Sentence: The shark swam towards the unsuspecting surfer and bit his leg off, causing a commotion on the beach.

2. PSG Tree:



Definitions:
S: Simple Declarative Clause
NP: Noun Phrase
VP: Verb Phrase
PP: Prepositional Phrase
PRT: Particle

3. Dependency Parse:


punct
conj
nmod
det
compound
case
det
amod
cc
dep
punct
compound:prt
obj
nmod:poss
obj
det
nmod
case
det
The shark swam towards the unsuspecting surfer and bit his leg off, causing a commotion on the beach.

Definitions:

Det: determiner
Compound: Compound
Punct: punctuation
Conj: conjunction
Nmod: noun modifier
Amod: adjectival modifier
CC: coordination
Dep: dependent
Compound:prt: phrasal verb particle compound
Obj: object
Nmod:poss: possessive noun modifier

4. SRL Parse:

Predicate: swam
Arguments:
- Arg0: The shark
- Arg1: the unsuspecting surfer
- Arg4: bit his leg off

Modifiers:
- DIR: “towards” - motion along a path (directional)

Predicate: causing
Arguments:
- Arg4: a commotion

Modifiers:
- LOC: “on the beach” - Where the action took place (location)

5. Pros and Cons of Each Parse Type

The three parse types used in this exercise each have their respective advantages and disadvantages when compared to the others. Each conveys different information from the others, and on their own are not as useful as they are when combined.

The PSG parse type is very useful in breaking down a sentence into very small parts and labeling it thoroughly while still conveying information about the sentence structure and word usage in the sentence. This can be useful for a machine to break down the structure and understand the sentence and the role of each word token in the sentence but can get to be very large on a lengthy or complex input sentence.

The dependency parse type was able to give a better overview of what each word did and how it related to the words around it in the sentence. This would also be useful for machines to understand what each word does in the context of the sentence around it, but may also generate too much and fairly repetitive output for some sentence structures, such as a complex sentence with a lot of punctuation.

Lastly, the SRL parse was the simplest and most straightforward parse type of the three. It used the predicate of the sentence, which is arguably the most important part, and returned a list of arguments and modifiers on that predicate. The simple and straightforward categorization of arguments makes this a good choice for a human to understand what the sentence is doing, and can be sufficient for some NLP applications that are not very nuanced. However, its strength can also be seen as a weakness: it extracts too little information out of text to be useful in a robust machine-learning model.