1. **Write a fairly complex sentence. The sentence should have at least 12 tokens. More points are awarded for sentences with more than one clause.**

Quote from *Can’t Hurt Me,* by David Goggins:

* “It won't always go your way, so you can't get trapped in this idea that just because you've imagined a possibility for yourself that you somehow deserve it.”
* To not have contractions, I extended the applicable words.

1. **Hand draw (no copy/paste) a PSG tree of the sentence, labeling POS. Briefly define all phrase terms that appear such as: S, SBAR, NP, VP, PP, etc. This resource may be helpful: https://gist.github.com/nlothian/9240750**

**S:** Simple declarative clause, i.e. one that is not introduced by a (possible empty) subordinating conjunction or a wh-word and that does not exhibit subject-verb inversion.

**SBAR:** Clause introduced by a (possibly empty) subordinating conjunction.

**ADVP:** Adverb Phrase

**NP:** Noun Phrase

**PP:** Prepositional Phrase

**VP:** Verb Phrase

**DT:** Determiner

**IN:** Preposition or subordinating conjunction

**MD:** Modal

**NN:** Noun

**POS:** Part of Speech

**PRP:** Personal pronoun

**PRP$:** Possessive pronoun

**RB:** Adverb

**VBN:** Verb, past participle

**VBP:** Verb, non-3rd person singular present

**Diagram, schematic

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

1. **Hand draw (no copy/paste) a dependency parse of the sentence, labeling dependency relations (but not POS). Briefly define all dependency relations that occurred in the parse. You can use the Stanford Dependencies Manual that is uploaded to Piazza.**

**ADVMOD:** adverb modifier; modifies the meaning of a word.

**AUX:** auxiliary; a non-main verb of a clause

**DEP:** dependent; used when system is unable to determine a more precise dependency relation between to words.

**DET:** determiner; the relation between the head of an NP and its determiner.

**DOBJ:** direct object; the direct object of a VP is the noun phrase which is the (accusative) object of the verb.

**MARK:** marker; the word introducing a finite clause subordinate to another clause.

**NEG:** negation modifier; the relation between a negation word and the word it modifies.

**NPADVMOD:** noun phrase as adverbial modifier; captures various places where something syntactically a NP is used as an adverbial modifier in a sentence.

**NSUBJ:** nominal subject; a noun phrase which is the syntactic subject of a clause.

**POBJ:** Object of a preposition; head of a noun phrase following the preposition, or the adverbs “here” and “there”. In the case of preposition stranding, the object can precede the preposition.

**PREP:** prepositional modifier; a prep of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition.

**PUNCT:** punctuation; used for any piece of punctuation in a clause. By default, punctuation is not retained in the output.

**RCMOD:** relative clause modifier; a rcmod of an NP is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause.

**Diagram, schematic

Description automatically generated**

1. **For the SRL parse, list the predicate, all arguments (numbered) and modifiers (TMP, LOC, etc.) for each verb in the sentence. Briefly discuss the numbered arguments and their relation to each verb. List each modifier and briefly define what it is. Refer to the list of arguments in the class slides.**

**Arg0:** the agent of the sentence

**Arg1:** the passive actor

**Arg2:** the ‘instrument’

**C-Arg1:** continuation argument

***Modifiers***

Argm-mod: modal; used to identify specific words that enable us to identify our rules

Argm-tmp: When the action happened

Argm-neg: negation

Argm-mnr: manner; How the action happened

**Text, letter

Description automatically generated**

**Text

Description automatically generated**

1. **Write a paragraph briefly summarizing the pros/cons of each parse type (your opinion) for your sample sentence.**

* The PSG tree seems to give the most detail. It has a tag for each word. The downside is that it creates a really deep tree. This could be cumbersome with a more complex sentence. The dependency tree seemed suitable for context models since you can see what words are needed for others, to illustrate a scene and the tree was much nicer to work with. Still, a tree is created and could lead to a deep tree. The PSG shows the parts of a sentence, and the dependency tree shows hoe they are held together. The SRL parse was easiest to work with and broke the sentence up into frames. This could be a lot to deal with if high numbers of frames were created. The SRL also helps with context because the modifiers tell you how arguments are used.