1. Ellipsis (…) is tagged as comma (,) because there are only 9 punctuations that are well tagged in collection of pos tags, and among them are #, $, (,), closing quote, opening quote, period, colon, and comma. The lack of tags for special characters may affect the pipeline’s functioning, for example: adding ellipsis in a sentence to emphasize the hesitation of the subject can cause the position change of dependency parse of punctuation; however, the extent of severeness seems to be trivial in case we don’t concentrate on punctuations much.

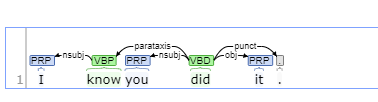


Figure 1. dependencies of sentence: “I know you did it” (https://corenlp.run/)

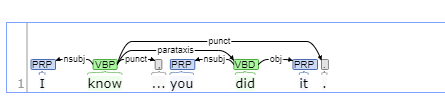


Figure 2. dependencies of sentence: “I know … you did it” (https://corenlp.run/)

1. NLP pipeline fails to identify an adverb in some cases that an adverb is composited by more than one words, for example: “in the least“ is an adverb, but after it goes through our NLP pipeline, it is considered as a prepositional phrase with a noun phrase inside. This situation is caused by the mandatory execution order of splitting and tagging. That is, we have to split our sentence first before we can assign pos tags to tokens when using CoreNLP, but once the words are separated, the original meaning may distort in the NLP pipeline.
2. NLP pipeline cannot do well when facing the mix of languages, that is for example, an English pipeline cannot correctly tag/annotate French phrase inside the sentence. Since the mix of uses of different languages can be very common among translation-related contents, linguistic-related contents, and bilingual users, if we easily generalize the use of this pipeline to all text corpus without a good knowledge of the content of corpus, this can be a severe issue.
3. Pos tag cannot classify the subordinating conjunctions and prepositions since they share the same tag which is “IN”, so the pipeline can fail when extracting information about them.
4. Pipeline cannot correctly identify a full stop, for example, Mr.s.: the second period is considered a full stop in this case, but in fact, it is not. It is not a proper solution if we remove all periods in text corpus before we process them, because that will brutally merge 2 separated sentences into one in some cases, so by observing the corpus and the outputted result, I found there is only one word that involves this problem, thus I replaced all “Mr.s.” with “Mrs” before scripting, and the problem is well solved.

p.s.:

there is also an issue with the WordNet, that is, some words are missing in WordNet; for example: relock is well recorded in many dictionaries including Collin, Free Dictionary, Lexico, etc., but WordNet returns null for this word.