Unit 6 Assignment

**ANLY:520-51 (Fall 2016)**

Dean D’souza

H.U. ID: 168424

# **Solutions:**

1. **The IOB format categorizes tagged tokens as I, O and B. Why are three tags necessary? What problem would be caused if we used I and O tags exclusively?**

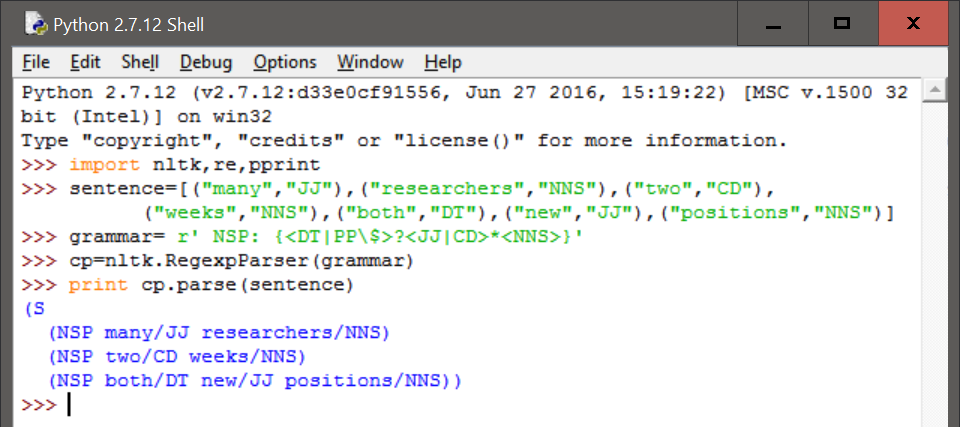
The IOB format, which categorizes tokens as “I” for inside the chunk, “O” for outside the chunk and “B” for beginning of the chunk, needs the three tags to effectively represent the constituents of the individual chunks. Tokens tagged with “B” helps to easily recognize where a chunk begins so that the user as well as the program can appropriately distinguish between different chunks (provided that the chunks do not overlap).

If only “I” and “O” tags are used, then the user and the program would be unable to distinguish between different chunks and would consider all tokens marked as “I” to belong to a single large chunk.

1. **Write a tag pattern to match noun phrases containing plural head nouns, e.g. "many/JJ researchers/NNS", "two/CD weeks/NNS", "both/DT new/JJ positions/NNS". Try to do this by generalizing the tag pattern that handled singular noun phrases.**

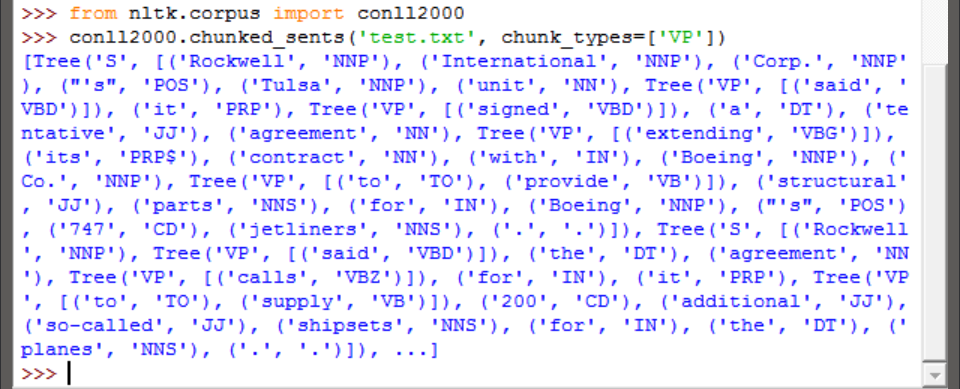
The tag pattern would be of the form {<DT|PP\$>?<JJ|CD>\*<NNS>}.

We test it out as follows:

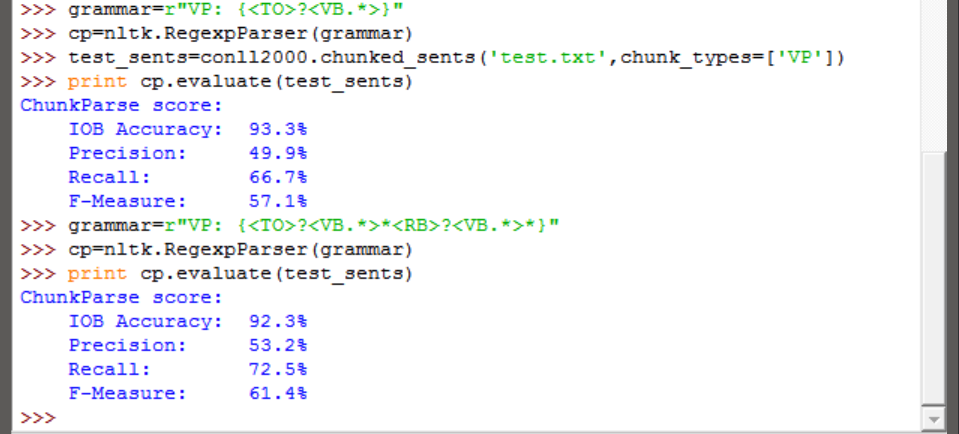


1. **Pick one of the three chunk types in the CoNLL corpus. Inspect the CoNLL corpus and try to observe any patterns in the POS tag sequences that make up this kind of chunk. Develop a simple chunker using the regular expression chunker nltk.RegexpParser. Discuss any tag sequences that are difficult to chunk reliably.**

We observe the ‘VP’ chunk type of the conll2000 corpus as follows:



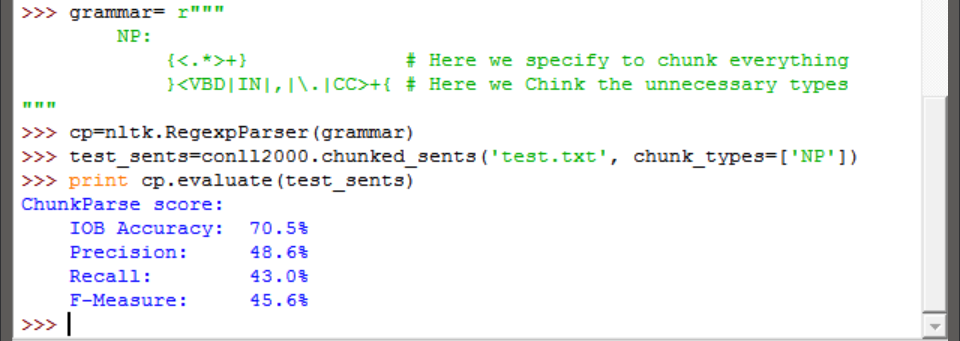
We develop the simple chunker as follows:



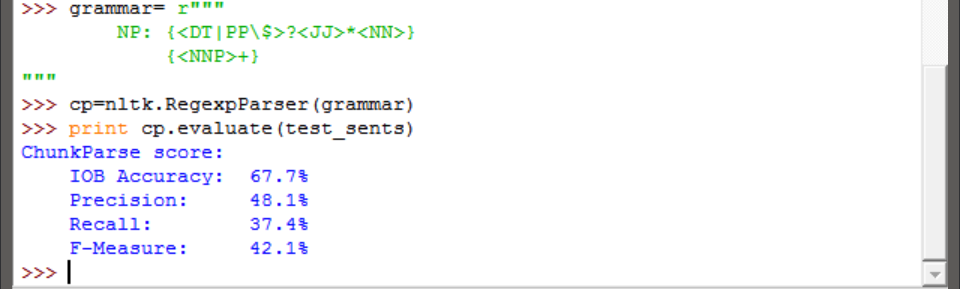
As we can see above, we attempt to create a good regular expression for capturing verb phrases but only end up with a precision of 53.2%. Further investigation of Verb Phrase makeups would give a better idea of how to construct a good regular expression for the same.

1. **An early definition of chunk was the material that occurs between chinks. Develop a chunker that starts by putting the whole sentence in a single chunk, and then does the rest of its work solely by chinking. Determine which tags (or tag sequences) are most likely to make up chinks with the help of your own utility program. Compare the performance and simplicity of this approach relative to a chunker based entirely on chunk rules.**

We create the required chunker for Noun Phrases (we also use conll2000 once again to evaluate) as follows:



We also test a simple gramma of two rules to the same test\_sents set as follows:



As we can see above, while the chinking based chunkers precision was higher, we can’t be absolutely sure that it is better than creating a grammar with all possible rules of Noun Phrases. However, it is easier to create and has more chances of capturing all Noun Phrases as compared to a chunker with a few rules for noun phrases.

1. **Write a tag pattern to cover noun phrases that contain gerunds, e.g. "the/DT receiving/VBG end/NN", "assistant/NN managing/VBG editor/NN". Add these patterns to the grammar, one per line. Test your work using some tagged sentences of your own devising.**

The tag patterns are shown as follows:



We test it out as follows:

