One of the more powerful aspects of the NLTK module is the Part of Speech tagging that it can do for you. This means labeling words in a sentence as nouns, adjectives, verbs...etc. Even more impressive, it also labels by tense, and more. Here's a list of the tags, what they mean, and some examples:

POS tag list:

CC coordinating conjunction

CD cardinal digit

DT determiner

EX existential there (like: "there is" ... think of it like "there exists")

FW foreign word

IN preposition/subordinating conjunction

JJ adjective 'big'

JJR adjective, comparative 'bigger'

JJS adjective, superlative 'biggest'

LS list marker 1)

MD modal could, will

NN noun, singular 'desk'

NNS noun plural 'desks'

NNP proper noun, singular 'Harrison'

NNPS proper noun, plural 'Americans'

PDT predeterminer 'all the kids'

POS possessive ending parent's

PRP personal pronoun I, he, she

PRP$ possessive pronoun my, his, hers

RB adverb very, silently,

RBR adverb, comparative better

RBS adverb, superlative best

RP particle give up

TO to go 'to' the store.

UH interjection errrrrrrrm

VB verb, base form take

VBD verb, past tense took

VBG verb, gerund/present participle taking

VBN verb, past participle taken

VBP verb, sing. present, non-3d take

VBZ verb, 3rd person sing. present takes

WDT wh-determiner which

WP wh-pronoun who, what

WP$ possessive wh-pronoun whose

WRB wh-abverb where, when

How might we use this? While we're at it, we're going to cover a new sentence tokenizer, called the PunktSentenceTokenizer. This tokenizer is capable of unsupervised machine learning, so you can actually train it on any body of text that you use. First, let's get some imports out of the way that we're going to use:

import nltk

from nltk.corpus import state\_union

from nltk.tokenize import PunktSentenceTokenizer

Now, let's create our training and testing data:

train\_text = state\_union.raw("2005-GWBush.txt")

sample\_text = state\_union.raw("2006-GWBush.txt")

One is a State of the Union address from 2005, and the other is from 2006 from past President George W. Bush.

Next, we can train the Punkt tokenizer like:

custom\_sent\_tokenizer = PunktSentenceTokenizer(train\_text)

Then we can actually tokenize, using:

tokenized = custom\_sent\_tokenizer.tokenize(sample\_text)

Now we can finish up this part of speech tagging script by creating a function that will run through and tag all of the parts of speech per sentence like so:

def process\_content():

try:

for i in tokenized[:5]:

words = nltk.word\_tokenize(i)

tagged = nltk.pos\_tag(words)

print(tagged)

except Exception as e:

print(str(e))

process\_content()

The output should be a list of tuples, where the first element in the tuple is the word, and the second is the part of speech tag. It should look like:

[('PRESIDENT', 'NNP'), ('GEORGE', 'NNP'), ('W.', 'NNP'), ('BUSH', 'NNP'), ("'S", 'POS'), ('ADDRESS', 'NNP'), ('BEFORE', 'NNP'), ('A', 'NNP'), ('JOINT', 'NNP'), ('SESSION', 'NNP'), ('OF', 'NNP'), ('THE', 'NNP'), ('CONGRESS', 'NNP'), ('ON', 'NNP'), ('THE', 'NNP'), ('STATE', 'NNP'), ('OF', 'NNP'), ('THE', 'NNP'), ('UNION', 'NNP'), ('January', 'NNP'), ('31', 'CD'), (',', ','), ('2006', 'CD'), ('THE', 'DT'), ('PRESIDENT', 'NNP'), (':', ':'), ('Thank', 'NNP'), ('you', 'PRP'), ('all', 'DT'), ('.', '.')] [('Mr.', 'NNP'), ('Speaker', 'NNP'), (',', ','), ('Vice', 'NNP'), ('President', 'NNP'), ('Cheney', 'NNP'), (',', ','), ('members', 'NNS'), ('of', 'IN'), ('Congress', 'NNP'), (',', ','), ('members', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('Supreme', 'NNP'), ('Court', 'NNP'), ('and', 'CC'), ('diplomatic', 'JJ'), ('corps', 'NNS'), (',', ','), ('distinguished', 'VBD'), ('guests', 'NNS'), (',', ','), ('and', 'CC'), ('fellow', 'JJ'), ('citizens', 'NNS'), (':', ':'), ('Today', 'NN'), ('our', 'PRP$'), ('nation', 'NN'), ('lost', 'VBD'), ('a', 'DT'), ('beloved', 'VBN'), (',', ','), ('graceful', 'JJ'), (',', ','), ('courageous', 'JJ'), ('woman', 'NN'), ('who', 'WP'), ('called', 'VBN'), ('America', 'NNP'), ('to', 'TO'), ('its', 'PRP$'), ('founding', 'NN'), ('ideals', 'NNS'), ('and', 'CC'), ('carried', 'VBD'), ('on', 'IN'), ('a', 'DT'), ('noble', 'JJ'), ('dream', 'NN'), ('.', '.')] [('Tonight', 'NNP'), ('we', 'PRP'), ('are', 'VBP'), ('comforted', 'VBN'), ('by', 'IN'), ('the', 'DT'), ('hope', 'NN'), ('of', 'IN'), ('a', 'DT'), ('glad', 'NN'), ('reunion', 'NN'), ('with', 'IN'), ('the', 'DT'), ('husband', 'NN'), ('who', 'WP'), ('was', 'VBD'), ('taken', 'VBN'), ('so', 'RB'), ('long', 'RB'), ('ago', 'RB'), (',', ','), ('and', 'CC'), ('we', 'PRP'), ('are', 'VBP'), ('grateful', 'JJ'), ('for', 'IN'), ('the', 'DT'), ('good', 'NN'), ('life', 'NN'), ('of', 'IN'), ('Coretta', 'NNP'), ('Scott', 'NNP'), ('King', 'NNP'), ('.', '.')] [('(', 'NN'), ('Applause', 'NNP'), ('.', '.'), (')', ':')] [('President', 'NNP'), ('George', 'NNP'), ('W.', 'NNP'), ('Bush', 'NNP'), ('reacts', 'VBZ'), ('to', 'TO'), ('applause', 'VB'), ('during', 'IN'), ('his', 'PRP$'), ('State', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('Union', 'NNP'), ('Address', 'NNP'), ('at', 'IN'), ('the', 'DT'), ('Capitol', 'NNP'), (',', ','), ('Tuesday', 'NNP'), (',', ','), ('Jan', 'NNP'), ('.', '.')]

At this point, we can begin to derive meaning, but there is still some work to do.