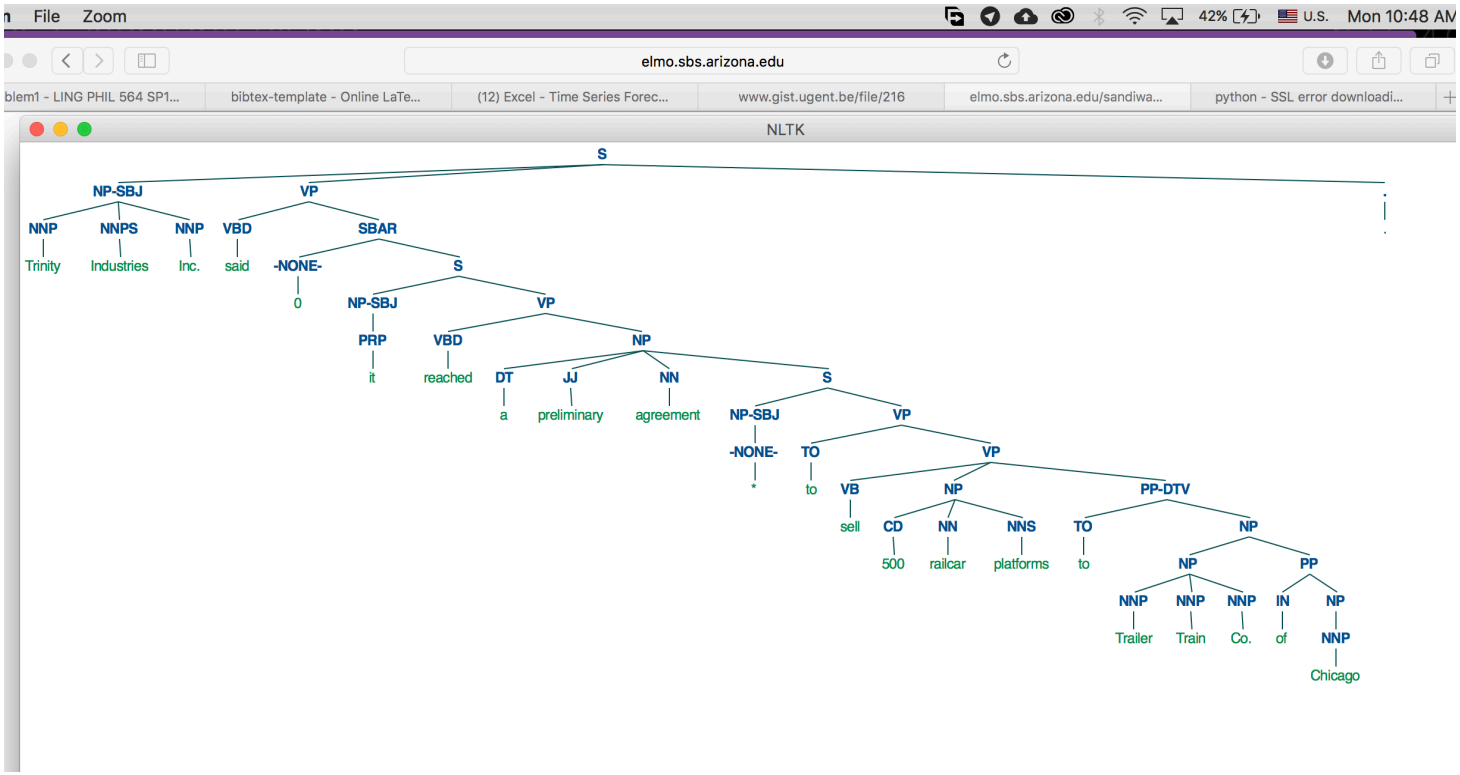
LING583 – Ragheb Al-Ghezi
Raghebalghezi@email.arizona.edu

Part1

LING583 – Ragheb Al-Ghezi
Raghebalghezi@email.arizona.edu

Part2



LING583 – Ragheb Al-Ghezi
Raghebalghezi@email.arizona.edu

Part3:

Analysis:
Despite the difficulty of the sentence, the POS tagger has done a quite good job in tagging the words. However, some words have been improperly tagged. For example, the tagger misclassifies the word "And" with the tag "RB", when it should label it with "CC". This misclassification is very predictable as the number of sentences in English starting with coordinating conjunction is rare, it is commonly advised against in writing practices, compared to those starting with Noun Phrases (and its components). In other words, the probability of their occurrences is low. Thus, a probabilistic POS tagger would not be in their favor. Another improper tagging in the sentence is with the word "or" that is not assigned any tag due to the existence of a comma concatenated with its beginning. The word tokenizer does not tokenize the string correctly. Finally, the word 'Asleep' is assigned the tag 'RB' (Adverb) when "JJ" is.

```python
# -*- coding: utf-8 -*-
import random
random.seed()


import nltk

raw=nltk.corpus.gutenberg.raw('carroll-alice.txt')

sents=nltk.sent_tokenize(raw)

rand_sent = random.randrange(1,len(sents))

words = nltk.tokenize.word_tokenize(sents[rand_sent])

sent_pos = nltk.pos_tag(words)

print(rand_sent) #I have chosen sent. #865 for POS accuracy analysis
print(sent_pos)
print(sents[865])

#865
#[("'And", 'RB'), ('be', 'VB'), ('quick', 'JJ'), ('about', 'IN'), ('it', 'PRP'),
# (',', ','), ("'", "''"), ('added', 'VBD'), ('the', 'DT'), ('Hatter', 'NNP'),
# (',', ','), ("'or", "''"), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'),
# ('asleep', 'RB'), ('again', 'RB'), ('before', 'IN'), ('it', 'PRP'), ("'s", 'VBZ')
# ('done', 'VBN'), ('.', '.'), ("'", "''")]

print(sents[865])
#'And be quick about it,' added the Hatter, 'or you'll be asleep again
#before it's done.'

```