Assignment #5

Tag POS and NER to the paragraph with different color

- All nouns = red
- All verbs = green
- All person tagged words = yellow
- All organization tagged words = blue
- All GPE tagged words = green

CODE

```
from nltk.tokenize import word_tokenize
from nltk import pos_tag, ne_chunk, RegexpParser
from nltk.chunk import conlltags2tree, tree2conlltags
import re
from termcolor import colored

text = "The family of Samsung Electronics chairman Lee KunHee will pay more th
an 12trn won ($10.78bn) in inheritance taxes on his estate. South Korea has on
e of the world's highest inheritance tax rates.Mr Lee, who is credited with tr
ansforming Samsung into a global electronics giant, died in October last year.
 The tax issue has been closely watched by investors as it could have affected
 the Lee family's stake in Samsung."
# text = "Tin and Ton are employee at IBM and live in Thailand"

def find_pos_tag (data):
    word = word_tokenize(data)
    word = pos_tag(word)
    return word

def capital_check(data):
    if(data[0].isupper()):
        for i in range (1,len(data)-1):
            if(data[i].isupper()):
                return False
        return True
    else:
        return False

def color_pos_tag (data):
    color_sentence = ""
    for i in data:
        if re.match(r"[N]+",i[1]):
            color_sentence += colored(i[0],'red') +" "
```

```
        elif re.match(r"[V]+",i[1]):
            color_sentence += colored(i[0],'green') +" "
        else:
            color_sentence += i[0] +" "
    return color_sentence


def color_NER(data):
    color_NER = ""
    for i in data:
        if (re.match("(NNP)",i[1])):
            if(re.match(".*(PERSON)",i[2])):
                    color_NER += colored(i[0],'yellow') +" "
            elif(re.match(".*(GPE)",i[2])):
                color_NER += colored(i[0],'green') +" "
            elif(re.match(".*(ORG).*",i[2])):
                color_NER += colored(i[0],'blue') +" "
        else:
                color_NER += i[0] +" "
    return color_NER

word_pos_tag = find_pos_tag(text)
print(f"POS:\n{color_pos_tag(word_pos_tag)}")

chunking_sentence = ne_chunk(word_pos_tag)

tree = tree2conlltags(chunking_sentence)
print("tree = ",tree)
print(f"\nNER:\n{color_NER(tree)}")
```

Result

```
D:\NLP>c:/python38/python.exe d:/NLP/5.py
POS:
The family of Samsung Electronics chairman Lee KunHee will pay more than 12trn won ( $ 10.78bn ) in inheritance taxes on his estate . South Korea has
 one of the world 's highest inheritance tax rates.Mr Lee , who is credited with transforming Samsung into a global electronics giant , died in Octob
er last year . The tax issue has been closely watched by investors as it could have affected the Lee family 's stake in Samsung .

NER:
The family of Samsung Electronics chairman Lee KunHee will pay more than 12trn won ( $ 10.78bn ) in inheritance taxes on his estate . South Korea has
 one of the world 's highest inheritance tax rates.Mr Lee , who is credited with transforming Samsung into a global electronics giant , died in last
year . The tax issue has been closely watched by investors as it could have affected the Lee family 's stake in Samsung .
```

```
tree =  [('The', 'DT', 'O'), ('family', 'NN', 'O'), ('of', 'IN', 'O'), ('Samsung', 'NNP', 'B-ORGANIZATION'), ('Electronics', 'NNP', 'I-ORGANIZATION')
, ('chairman', 'NN', 'O'), ('Lee', 'NNP', 'B-PERSON'), ('KunHee', 'NNP', 'I-PERSON'), ('will', 'MD', 'O'), ('pay', 'VB', 'O'), ('more', 'JJR', 'O'),
('than', 'IN', 'O'), ('12trn', 'CD', 'O'), ('won', 'VBD', 'O'), ('(', '(', 'O'), ('$', '$', 'O'), ('10.78bn', 'CD', 'O'), (')', ')', 'O'), ('in', 'IN
', 'O'), ('inheritance', 'NN', 'O'), ('taxes', 'NNS', 'O'), ('on', 'IN', 'O'), ('his', 'PRP$', 'O'), ('estate', 'NN', 'O'), ('.', '.', 'O'), ('South'
, 'NNP', 'B-GPE'), ('Korea', 'NNP', 'I-GPE'), ('has', 'VBZ', 'O'), ('one', 'CD', 'O'), ('of', 'IN', 'O'), ('the', 'DT', 'O'), ('world', 'NN', 'O'), (
"'s", 'POS', 'O'), ('highest', 'JJS', 'O'), ('inheritance', 'NN', 'O'), ('tax', 'NN', 'O'), ('rates.Mr', 'NN', 'O'), ('Lee', 'NNP', 'B-PERSON'), (','
, ',', 'O'), ('who', 'WP', 'O'), ('is', 'VBZ', 'O'), ('credited', 'VBN', 'O'), ('with', 'IN', 'O'), ('transforming', 'VBG', 'O'), ('Samsung', 'NNP',
'B-PERSON'), ('into', 'IN', 'O'), ('a', 'DT', 'O'), ('global', 'JJ', 'O'), ('electronics', 'NN', 'O'), ('giant', 'NN', 'O'), (',', ',', 'O'), ('died'
, 'VBD', 'O'), ('in', 'IN', 'O'), ('October', 'NNP', 'O'), ('last', 'JJ', 'O'), ('year', 'NN', 'O'), ('.', '.', 'O'), ('The', 'DT', 'O'), ('tax', 'NN
', 'O'), ('issue', 'NN', 'O'), ('has', 'VBZ', 'O'), ('been', 'VBN', 'O'), ('closely', 'RB', 'O'), ('watched', 'VBN', 'O'), ('by', 'IN', 'O'), ('inves
tors', 'NNS', 'O'), ('as', 'IN', 'O'), ('it', 'PRP', 'O'), ('could', 'MD', 'O'), ('have', 'VB', 'O'), ('affected', 'VBN', 'O'), ('the', 'DT', 'O'), (
'Lee', 'NNP', 'B-ORGANIZATION'), ('family', 'NN', 'O'), ("'s", 'POS', 'O'), ('stake', 'NN', 'O'), ('in', 'IN', 'O'), ('Samsung', 'NNP', 'B-GPE'), ('.
', '.', 'O')]
```