

# NeuralCoref & OpenIE

May 12, 2021

## 0.1 Environment Setup

Anaconda Environment via Ubuntu 20.04

- Jupyter Lab
- python 3.8 : spacy, neuralcoref, & StanfordOpenIE
- Java8

\* In-depth instructions for setting up the environment can be found [in this gist](#).

---

## 1 Creating a NeuralCoref Function

```
[1]: import spacy
import neuralcoref

def NeuralCoref(text, visualize=False, debug=False):
    nlp = spacy.load('en')
    neuralcoref.add_to_pipe(nlp)

    doc = nlp(text)

    if visualize: print("=====> INPUT_
↪<=====\n\n%s" % text);

    for i in range(len(doc._.coref_clusters)):
        a = doc._.coref_clusters[i].mentions[-1]
        b = doc._.coref_clusters[i].mentions[-1]._.coref_cluster.main
        text = text.replace(str(a), str(b))
        if debug: print("|- ", text);
    if visualize: print("\n\n=====> OUTPUT_
↪<=====\n\n%s" % text);

    return text

[2]: # Begin with a simple 1 sentence string.
str_1 = 'John have dinner today and he enjoyed it.'
```

```
# Replace NeuralCoref parameter with desired string
a = NeuralCoref(str_1, visualize=True)
```

=====> INPUT <=====

John have dinner today and he enjoyed it.

=====> OUTPUT <=====

John have dinner today and John enjoyed it.

---

## Compatibility Issues

- Inconsistent output given when running on a Windows Machine Vs. Linux System

```
[3]: # Using a different sentence
str_2 = 'My sister has a dog. She loves him.'

# Obtain clusters for string 2 (on linux system)
print("=====> Linux Machine <=====")
linux = NeuralCoref(str_2, debug=True)
```

=====> Linux Machine <=====

| - My sister has a dog. My sister loves him.  
| - My sister has a dog. My sister loves a dog.

=====> Windows Machine <=====

| - My sister has a dog. My sister loves a dog.

---

## 2 Create Custom OpenIE Function

```
[4]: from openie import StanfordOpenIE

def OpenIE(text, visualize=False, debug=False):
    with StanfordOpenIE() as client:
        i=1
        if visualize: print("=====> INPUT_
↪<=====\n\n%s" % text);
        if visualize: print("\n\n=====> OUTPUT_
↪<=====");
        for triple in client.annotate(text):
            if i: i=0; print();
            print("==> ", triple)
```

```
[5]: # Returning back our simple 1 sentence string
str_1 = 'John have dinner today and he enjoyed it.'

# Replace OpenIE parameter with desired string
b = OpenIE(str_1, visualize=True)
```

```
=====> INPUT <=====
```

```
John have dinner today and he enjoyed it.
```

```
=====> OUTPUT <=====
```

```
Starting server with command: java -Xmx8G -cp
/home/blurry/.stanfordnlp_resources/stanford-corenlp-4.1.0/*
edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 9000 -timeout 60000
-threads 5 -maxCharLength 100000 -quiet True -serverProperties
corenlp_server-2f3872ca8ba642b4.props -preload openie
```

```
==> {'subject': 'he', 'relation': 'enjoyed', 'object': 'it'}
==> {'subject': 'John', 'relation': 'have dinner at_time', 'object': 'today'}
```

## 2.1 OpenIE Example using Java as shown on [OpenIE's Website](#)

```
[6]: %%%bash
echo -e "=====> INPUT <=====\\n"; cat ex.txt

# java -cp "stanford-corenlp-4.2.0/*" -Xmx5g edu.stanford.nlp.pipeline.
↪StanfordCoreNLP -file ex.txt;

echo -e "\\n\\n=====> OUTPUT <=====\\n"; cat ex.
↪txt.out
```

```
=====> INPUT <=====
```

John have dinner today and he enjoyed it.

=====> OUTPUT <=====

Document: ID=ex.txt (1 sentences, 9 tokens)

Sentence #1 (9 tokens):

John have dinner today and he enjoyed it.

Tokens:

[Text=John CharacterOffsetBegin=0 CharacterOffsetEnd=4 PartOfSpeech=NNP  
Lemma=John NamedEntityType=PERSON]  
[Text=have CharacterOffsetBegin=5 CharacterOffsetEnd=9 PartOfSpeech=VBP  
Lemma=have NamedEntityType=0]  
[Text=dinner CharacterOffsetBegin=10 CharacterOffsetEnd=16 PartOfSpeech=NN  
Lemma=dinner NamedEntityType=0]  
[Text=today CharacterOffsetBegin=17 CharacterOffsetEnd=22 PartOfSpeech=NN  
Lemma=today NamedEntityType=DATE NormalizedNamedEntityType=THIS P1D Timex=<TIMEX3  
alt\_value="THIS P1D" anchorTimeID="t0" temporalFunction="true" tid="t1"  
type="DATE" valueFromFunction="tf0">today</TIMEX3>]  
[Text=and CharacterOffsetBegin=23 CharacterOffsetEnd=26 PartOfSpeech=CC  
Lemma=and NamedEntityType=0]  
[Text=he CharacterOffsetBegin=27 CharacterOffsetEnd=29 PartOfSpeech=PRP Lemma=he  
NamedEntityType=0]  
[Text=enjoyed CharacterOffsetBegin=30 CharacterOffsetEnd=37 PartOfSpeech=VBD  
Lemma=enjoy NamedEntityType=0]  
[Text=it CharacterOffsetBegin=38 CharacterOffsetEnd=40 PartOfSpeech=PRP Lemma=it  
NamedEntityType=0]  
[Text=. CharacterOffsetBegin=40 CharacterOffsetEnd=41 PartOfSpeech=. Lemma=.  
NamedEntityType=0]

Dependency Parse (enhanced plus plus dependencies):

root(ROOT-0, have-2)  
nsubj(have-2, John-1)  
obj(have-2, dinner-3)  
obl:tmod(have-2, today-4)  
cc(enjoyed-7, and-5)  
nsubj(enjoyed-7, he-6)  
conj:and(have-2, enjoyed-7)  
obj(enjoyed-7, it-8)  
punct(have-2, .-9)

Extracted the following NER entity mentions:

John	PERSON	PERSON:0.9861268552545607
today	DATE	DATE:-1.0
he	PERSON	-

Coreference set:

(1,6,[6,7]) -> (1,1,[1,2]), that is: "he" -> "John"

Coreference set:

(1,8,[8,9]) -> (1,3,[3,4]), that is: "it" -> "dinner"

---

### 3 Applying NeuralCoref before applying OpenIE

```
[7]: # Replace NeuralCoref parameter with desired string

# Step 1.) Run co-reference resolution on string using NeuralCoref.
nc = NeuralCoref(str_1)

# Step 2.) Run the output through OpenIE
c = OpenIE(nc, visualize=True)
```

=====> INPUT <=====

John have dinner today and John enjoyed it.

=====> OUTPUT <=====

```
Starting server with command: java -Xmx8G -cp
/home/blurry/.stanfordnlp_resources/stanford-corenlp-4.1.0/*
edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 9000 -timeout 60000
-threads 5 -maxCharLength 100000 -quiet True -serverProperties
corenlp_server-2129490e60394052.props -preload openie
```

```
==> {'subject': 'John', 'relation': 'enjoyed', 'object': 'it'}
==> {'subject': 'John', 'relation': 'have dinner at_time', 'object': 'today'}
```

---

#### 3.1 References

- [1] “neuralcoref · spaCy Universe,” neuralcoref, 2016. [Online]. Available: <https://spacy.io/universe/project/neuralcoref> [Accessed: 11-May-2021]
- [2] “The Stanford Natural Language Processing Group,” Stanford.edu, 2015. Available: <https://nlp.stanford.edu/software/openie.html> [Accessed May 11, 2021].