

/ cskg / ESWC2021 /

Name	Last Modified
Analyze CSKG.ipynb	2 minutes ago
CSKG Embeddings.ipynb	12 hours ago

CSKG.ipynb Playing with grounding.ipynb Analyze CSKG.ipynb

4 vCPU + 16 GiB Python 3 (Data Science) Share

Playing with CSKG grounding

Setup for grounding

```
conda create -n mowgli python=3.6
conda activate mowgli

git clone https://github.com/ucinlp/mowgli-uci

mv mowgli-uci grounding

cd grounding

pip install -r requirements.txt
conda install --yes faiss-cpu -c pytorch -n mowgli
python -m spacy download en_core_web_lg

cd ..
```

```
[30]: !apt-get update -y
!apt-get install -y graphviz libgraphviz-dev graphviz-dev pkg-config
!python -m pip install pygraphviz

Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://deb.debian.org/debian buster InRelease
Hit:3 http://deb.debian.org/debian buster-updates InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libgraphviz-dev' instead of 'graphviz-dev'
graphviz is already the newest version (2.40.1-6+deb10u1).
libgraphviz-dev is already the newest version (2.40.1-6+deb10u1).
pkg-config is already the newest version (0.29-6).
0 upgraded, 0 newly installed, 0 to remove and 34 not upgraded.
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Requirement already satisfied: pygraphviz in /opt/conda/lib/python3.7/site-packages (1.7)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

```
[31]: # IMPORTS
import pygraphviz as pgv
from IPython.display import Image
```

```
[32]: !ls ../grounding

LICENSE      graphify      groundcskg   requirements1.txt
README.md    graphify_documentation  pytest.ini   setup.py
demo         groundcn     requirements.txt  tests
```

I. Parsing questions and answers

```
[33]: import sys
sys.path.append('.')
sys.path.append('../grounding/')
from groundcn.graphify import graphify
```

```
-----
ModuleNotFoundError: Traceback (most recent call last)
<ipython-input-33-75a0c8a514> in <module>
      1 sys.path.append('.')
      2 sys.path.append('../grounding/')
----> 3 from groundcn.graphify import graphify

ModuleNotFoundError: No module named 'groundcn.graphify'
```

```
[34]: def draw(G):
return Image(G.draw(format='png', prog='dot'))
```

```
[35]: sentences=[
    'Max looked for the onions so that he could make a stew.',
    # 'To get the bathroom counters dry after washing your face, take a small hand Lotion and wipe away the extra water around the sink.',
    # 'To get the bathroom counters dry after washing your face, take a small hand towel and wipe away the extra water around the sink.'
]
```

```
[36]: parse_trees=graphify.graphify_dataset(sentences)
```

```
-----
NameError: Traceback (most recent call last)
<ipython-input-36-0f751a47034a> in <module>
----> 1 parse_trees=graphify.graphify_dataset(sentences)

NameError: name 'graphify' is not defined
```

```
[37]: parse_trees
```

```
-----
NameError: Traceback (most recent call last)
<ipython-input-37-f630f8bd11f0> in <module>
----> 1 parse_trees

NameError: name 'parse_trees' is not defined
```

Nicer visualization

```
[38]: rels=[]
for sent_data in parse_trees:
    #print('Sentence:', sent_data['sentence'])
    #print('Tokenized sentence', sent_data['tokenized_sentence'])

    G=pgv.AGraph(strict=False, directed=True)
```

```

nodes={}
for n_id, n_data in sent_data['nodes'].items():
    nodes[n_id]=' '.join(n_data['phrase'])

for e_id, e_data in sent_data['edges'].items():
    n1=nodes[e_data['head_node_id']]
    n2=nodes[e_data['tail_node_id']]
    rel=e_data['edge_name']
    edge=(e_data['head_node_id'], rel, e_data['tail_node_id'])
    rels.append(edge)

G.add_edge(n1, n2, label=rel)

```

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-38-b632e413b3f0> in <module>
      1 rels=[]
----> 2 for sent_data in parse_trees:
      3     #print('Sentence:', sent_data['sentence'])
      4     #print('Tokenized sentence', sent_data['tokenized_sentence'])
      5
NameError: name 'parse_trees' is not defined

```

[39]: draw(G)

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-39-3a7e7a753864> in <module>
----> 1 draw(G)
NameError: name 'G' is not defined

```

II. Grounding questions and questions to ConceptNet

[40]: `from groundcn.graphify import link`

```

-----
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-40-b197fa7b884c> in <module>
----> 1 from groundcn.graphify import link
ModuleNotFoundError: No module named 'groundcn.graphify'

```

Note: The Numberbatch file should be downloaded from [here](#), placed into the directory `../output/embeddings`, and `gunzip-ed`.

[41]: `numberbatch_file='../output/embeddings/numberbatch-en-19.08.txt'`

[42]: `linked_data=link.link(parse_trees, embedding_file=numberbatch_file)`

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-42-8c0d79c1886c> in <module>
----> 1 linked_data=link.link(parse_trees, embedding_file=numberbatch_file)
NameError: name 'link' is not defined

```

[43]: `rels`

[43]: `[]`

```

[44]: links={}
for sent_data in linked_data:
    print('Sentence:', sent_data['sentence'])
    linkedG=pgv.AGraph(strict=False, directed=True)
    for n_id, n_data in sent_data['nodes'].items():
        print('Node phrase:', n_data['phrase'])
        for c in reversed(n_data['candidates']):
            print(c)
            print()

            links[n_id]=list(reversed(n_data['candidates']))[0]['uri']

    for edge in rels:
        linkedG.add_edge(links[edge[0]], links[edge[2]], label=edge[1])
    print()

```

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-44-538087f38961> in <module>
      1 links={}
----> 2 for sent_data in linked_data:
      3     print('Sentence:', sent_data['sentence'])
      4     linkedG=pgv.AGraph(strict=False, directed=True)
      5     for n_id, n_data in sent_data['nodes'].items():
NameError: name 'linked_data' is not defined

```

[45]: `draw(linkedG)`

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-45-f178af979c31> in <module>
----> 1 draw(linkedG)
NameError: name 'linkedG' is not defined

```

III. Grounding to CSKG

[46]: `from groundcsgk.graphify import link`

```

-----
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-46-d64d3f9c725c> in <module>
----> 1 from groundcsgk.graphify import link
ModuleNotFoundError: No module named 'groundcsgk.graphify'

```

Note: The BERT embeddings file should be downloaded from [here](#), placed into the directory `../output/embeddings`, and `gunzip-ed`.

[47]: `#graph_emb_file='../output/embeddings/graph_embedding.tsv'`
`bert_file='../output/embeddings/bert_nli_large_w2v_format.txt'`

[48]: `import importlib`
`importlib.reload(link)`

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-48-75d21a497087> in <module>  
      1 import importlib  
----> 2 importlib.reload(link)  
  
NameError: name 'link' is not defined
```

```
[49]: linked_data=link.link(parse_trees, embedding_file=bert_file)
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-49-2189471b0561> in <module>  
----> 1 linked_data=link.link(parse_trees, embedding_file=bert_file)  
  
NameError: name 'link' is not defined
```

```
[50]: links={}  
for sent_data in linked_data:  
    print('Sentence:', sent_data['sentence'])  
    linkedG=pgv.AGraph(strict=False, directed=True)  
    for n_id, n_data in sent_data['nodes'].items():  
        print('Node phrase:', n_data['phrase'])  
        for c in reversed(n_data['candidates']):  
            print(c)  
            print()  
            links[n_id]=list(reversed(n_data['candidates']))+[0]['uri']  
  
    for edge in rels:  
        linkedG.add_edge(links[edge[0]], links[edge[2]], label=edge[1])  
    print()
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-50-188687932006> in <module>  
      1 links={}  
----> 2 for sent_data in linked_data:  
      3     print('Sentence:', sent_data['sentence'])  
      4     linkedG=pgv.AGraph(strict=False, directed=True)  
      5     for n_id, n_data in sent_data['nodes'].items():  
  
NameError: name 'linked_data' is not defined
```

```
[51]: draw(linkedG)
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-51-8270af039c93> in <module>  
----> 1 draw(linkedG)  
  
NameError: name 'linkedG' is not defined
```

```
[ ]:
```