

Reproducible Results on MIRC TFS

- ❖ **Source Code Author:** Sungmin Script for MIRC Data on Jupiter Python
- ❖ **Modified to Reproduce the results:** Pradeep for MIRC Data on raw python

Work done this week

- ❖ **Work on comments provided during 23June2017 meeting by Prof**
- ❖ **Familiarizing the Python code and results reproduced**
- ❖ **Interpreting results and starting with Report writing**
- ❖ **Connect with Sungmin to get KT on work done so far to reproduce cluster results**

Comments received from 23 June 2017 meeting

1. **How RadLex categories are analysed?**
2. **How word (frequency) encoded – TF/IDF or Entropy?**

How RadLex Categories are analyzed

❖ Code consider Term Frequency analysis for all categories

1. Document
2. History
3. Findings
4. Diagnosis
5. DDX
6. Discussion
7. Comments
8. References

```
# Read teaching files and grab all RadLex terms
def RSNA_parse3(url):

    conn_term = ""
    with urllib.request.urlopen(url) as url:
        sou = url.read()
        soup = BeautifulSoup(sou)

    global kv_pairs_all

    big_title = soup.find_all('h1')[0].text # title of TF
    title = soup.find_all('h2') # This is a title for each category (ddx, findi
    title[0] = 'Document'
    print(title)

    temp2 = soup.find_all('div', class_ = "hide")
```

```
['Document', <h2>Keywords</h2>, <h2>History</h2>,
<h2>Findings</h2>, <h2>Diagnosis</h2>, <h2>Differential</h2>,
<h2>Discussion</h2>, <h2>References</h2>]
['Document', <h2>Keywords</h2>, <h2>History</h2>,
<h2>Findings</h2>, <h2>Diagnosis</h2>, <h2>Differential</h2>,
<h2>Discussion</h2>, <h2>References</h2>]
['Document', <h2>Keywords</h2>, <h2>History</h2>,
<h2>Findings</h2>, <h2>Diagnosis</h2>, <h2>Differential</h2>,
<h2>Discussion</h2>, <h2>References</h2>]
```

In [7]:

How word(frequency) encoded?

❖ Word frequency is encoded using Entropy criteria

1. While building decision tree and
2. During Cluster Analysis

```
big_idx = 1
def BigClusterAnal(tf, tm): # tf = his_tf, ddx_tf, etc
    global big_idx
    membership = fcluster(Z, 45, criterion='maxclust')
    temp_tf = tf.T
    temp_tf['Membership'] = membership.tolist()

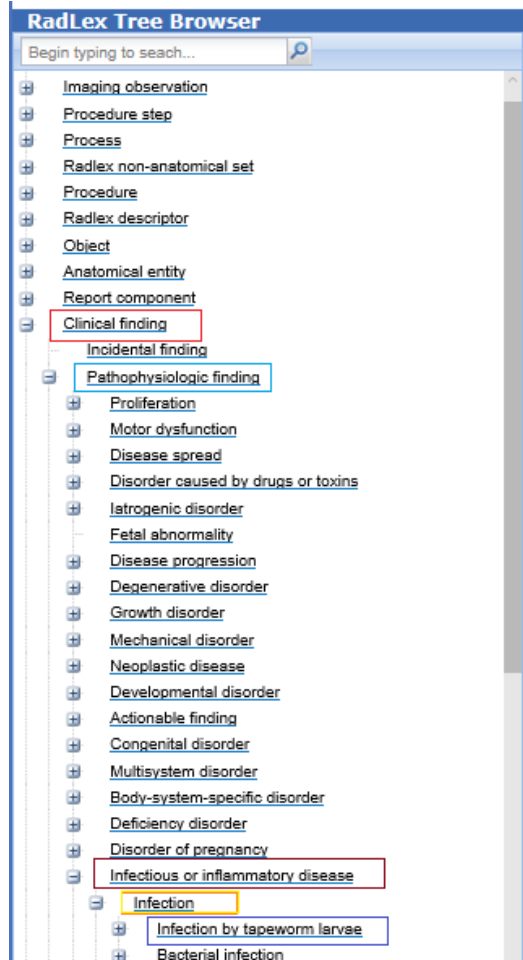
    # Re-index Membership
    temp_tf.Membership = tm

    # Build a decision tree
    treeclf = tree.DecisionTreeClassifier(criterion='entropy', min_samples_split=25)
    # Commented by Prady as min_impurity_split is not being accepted as a valid attribute
    #treeclf = tree.DecisionTreeClassifier(criterion='entropy', min_samples_split=25, min_impurity_split=0.00000002)
    treeclf = treeclf.fit(temp_tf.ix[:, temp_tf.columns != 'Membership'], temp_tf['Membership'])

    with open("haha_"+str(big_idx)+".dot", 'w') as f:
        f = tree.export_graphviz(treeclf, out_file=f)

    os.unlink("haha_"+str(big_idx)+".dot")
```


What I am working towards?



L1	L2	L3 +
Clinical finding	Pathophysiologic finding	<ul style="list-style-type: none"> ▪ Infectious or Inflammatory disease (16): Infection (5), Inflammation (4): <ul style="list-style-type: none"> - Rheumatic disease (3): Arthritis (2) <ul style="list-style-type: none"> - Enteritis (1) ▪ Proliferation (13): Focal Proliferation: Cyst (4), Neoplasm (8): <ul style="list-style-type: none"> - Neuroepitheliomatous neoplasm (4) - Blood Vessel Neoplasm (1) - <u>Myomatous neoplasm (1)</u> - Germ cell neoplasm: Non-seminomatous germ cell tumor (1) <ul style="list-style-type: none"> - Fatty neoplasm (1) ▪ Growth disorder (10): Aplasia (3), Dysplasia (1) ▪ Mechanical disorder (10): Flow disorder (5): Hemorrhage (1), Disorder caused by drugs or toxins (1), Displacement (2): <ul style="list-style-type: none"> - Displaced Substance: Displaced gas (1) - Malposition: Malalignment (1)

=====CLUSTER 12=====

['osteomyelitis => **infection** => **infectious or inflammatory disease** => **pathophysiologic finding** => **clinical finding** => ']
 ['infection => infectious or inflammatory disease => pathophysiologic finding => clinical finding => ']
 ['acute => temporal descriptor => Radlex descriptor => ']
 ['metaphysis => zone of long bone => zone of bone organ => organ zone => organ region => cardinal organ part => anatomical structure => material anatomical entity => anatomical entity => ']
 ['epiphysis => zone of long bone => zone of bone organ => organ zone => organ region => cardinal organ part => anatomical structure => material anatomical entity => anatomical entity => ']
 ['right => laterality => location descriptor => Radlex descriptor => ']
 ['adjacent => location descriptor => Radlex descriptor => ']

Work Completed: Understood the Py code and results

Work in Progress: Reproducing Summery Report

Work Planned for next two days:

1. Meeting Sungmin and obtaining all KT and Files
2. Work On Reports

Work Planned for Next Week:

1. Work towards the comments received from meeting

Thank You!