openav_notebook_idf

October 16, 2022

This is the notebook for tf-idf embeddings and visualizations

[18]:

```
# Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.manifold import TSNE
from sklearn.feature_extraction.text import TfidfVectorizer
```

[19]:

```
reports = pd.read_csv('open_ave_data.csv')
reports = reports.dropna()
reports.head(3)
```

[19]:

```
   Unnamed: 0                                         ReportText  \
0           0  EXAM: CHEST RADIOGRAPHY EXAM DATE: 06/01/2019 …
1           1  EXAM: CHEST RADIOGRAPHY EXAM DATE: 05/23/2020 …
2           2  EXAM: CHEST RADIOGRAPHY EXAM DATE: 12/13/2019 …

                                            findings  \
0  FINDINGS: Lungs/Pleura: No focal opacities evi…
1  FINDINGS: Lungs/Pleura: No focal opacities evi…
2  FINDINGS: Lungs/Pleura: No focal opacities evi…

                        clinicaldata  \
0       CLINICAL HISTORY: Cough. \n\n
1  CLINICAL HISTORY: CHEST PAIN. \n\n
2  CLINICAL HISTORY: CHEST PAIN. \n\n

                                            ExamName  \
0  EXAM: CHEST RADIOGRAPHY EXAM DATE: 06/01/2019 …
1  EXAM: CHEST RADIOGRAPHY EXAM DATE: 05/23/2020 …
2  EXAM: CHEST RADIOGRAPHY EXAM DATE: 12/13/2019 …

                                          impression
0      IMPRESSION: Normal 2-view chest radiography.
1  IMPRESSION: No acute cardiopulmonary abnormali…
2     IMPRESSION: No acute cardiopulmonary process.
```

```
[20]: report_findings = reports['findings'].str.split().tolist()
      report_clinicaldata = reports['clinicaldata'].str.split().tolist()
      report_examname = reports['ExamName'].str.split().tolist()
      report_impression = reports['impression'].str.split().tolist()
      # Take the limit to be the first tenth of values
      findings_limit = len(report_findings) * 0.1
      corpus_findings=[word for i in report_findings if isinstance(i, list) for word␣
       ↪in i ]
      corpus_clinicaldata=[word for i in report_clinicaldata if isinstance(i, list)␣
       ↪for word in i ]
      corpus_examname=[word for i in report_examname if isinstance(i, list) for word␣
       ↪in i ]
      corpus_impression=[word for i in report_impression if isinstance(i, list) for␣
       ↪word in i ]
```

```
[21]: vectorizerF=TfidfVectorizer()
      vectorizerC=TfidfVectorizer()
      vectorizerE=TfidfVectorizer()
      vectorizerI=TfidfVectorizer()
```

```
[22]: X_findings = vectorizerF.fit_transform(corpus_findings)
      X_clinicaldata = vectorizerC.fit_transform(corpus_clinicaldata)
      X_examname = vectorizerE.fit_transform(corpus_examname)
      X_impression = vectorizerI.fit_transform(corpus_impression)
```

```
[23]: vectorizerF.get_feature_names_out()
      vectorizerC.get_feature_names_out()
      vectorizerE.get_feature_names_out()
      vectorizerI.get_feature_names_out()
```

```
[23]: array(['00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10',
             '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '2014',
             '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022',
             '21', '22', '2249', '2251', '23', '24', '25', '26', '27', '28',
             '29', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39',
             '40', '41', '42', '43', '44', '45', '46', '47', '48', '49', '50',
             '51', '52', '53', '54', '55', '56', '57', '58', '59',
             '______________________________________', '_lcs1', '_lew2',
             'abnormalities', 'abnormality', 'above', 'active', 'acute',
             'aeration', 'agree', 'airspace', 'airway', 'airways', 'alveolar',
             'am', 'amount', 'an', 'and', 'answered', 'aortic', 'apex',
             'apical', 'apparatus', 'appear', 'appearance', 'appearing',
             'appropriate', 'approved', 'are', 'areas', 'artifact', 'as',
             'associated', 'asthma', 'at', 'atelectasis', 'atherosclerosis',
             'atrium', 'attending', 'attributable', 'atypical', 'authenticated',
             'base', 'bases', 'basilar', 'batch', 'be', 'been', 'below',
             'bhardwaj', 'bibasal', 'bibasilar', 'bilateral', 'bilaterally',
             'bonetti', 'borderline', 'both', 'bottom', 'bronchial',
             'bronchiolitis', 'bronchitic', 'bronchitis', 'but', 'by', 'call',
             'can', 'cardiac', 'cardiomediastinal', 'cardiomegaly',
             'cardiopulmonary', 'cardiothymic', 'carina', 'catheter', 'cava',
             'cc', 'cdt', 'central', 'change', 'changes', 'chest', 'chf',
             'chronic', 'clear', 'cm', 'code', 'combination', 'compared',
             'compatible', 'complete', 'compressive', 'concern', 'concerning',
             'conclusion', 'conclusions', 'congestion', 'congestive',
             'consistent', 'consolidation', 'consolidations', 'consolidative',
             'consultation', 'contusion', 'copd', 'corrected', 'could',
             'created', 'crowding', 'cst', 'cuffing', 'date', 'decompensation',
             'decrease', 'decreased', 'definite', 'demonstrated', 'densities',
             'density', 'described', 'development', 'devices', 'dictated',
             'dictatedtime', 'dictation', 'diffuse', 'diminished', 'disease',
             'distal', 'do', 'dr', 'drain', 'dt', 'due', 'edema', 'edited',
             'effusion', 'effusions', 'electronically', 'endotracheal', 'ends',
             'enlarged', 'enteric', 'et', 'etiology', 'ett', 'evidence', 'exam',
             'examination', 'extensive', 'extremity', 'failure', 'fax', 'field',
             'film', 'films', 'final', 'finalized', 'finding', 'findings',
             'fluid', 'focal', 'for', 'foundation', 'frequently', 'from',
             'further', 'ganz', 'greater', 'grossly', 'group', 'gs', 'hardware',
             'has', 'have', 'haziness', 'hazy', 'heart', 'hilar',
             'hyperinflation', 'i70', 'icd10', 'id', 'identified', 'iii', 'ij',
             'images', 'imaging', 'impression', 'impressions', 'improved',
             'improvement', 'in', 'increase', 'increased', 'infection',
             'infiltrate', 'infiltrates', 'inspiration', 'internal',
             'interpretation', 'interpreted', 'interstitial', 'interval',
             'interventional', 'intrathoracic', 'is', 'its', 'jugular', 'just',
             'key', 'large', 'layering', 'left', 'level', 'likely', 'limits',
             'line', 'linear', 'lines', 'lobar', 'lobe', 'lobes', 'located',
             'location', 'low', 'lower', 'lung', 'lungs', 'markings', 'mass',
             'may', 'md', 'medial', 'mediastinum', 'medical', 'mid', 'mild',
             'mildly', 'minimal', 'moderate', 'most', 'multifocal', 'name',
             'nasogastric', 'near', 'negative', 'new', 'no', 'nonspecific',
             'nor', 'normal', 'not', 'noted', 'of', 'on', 'one',
             'opacification', 'opacities', 'opacity', 'opportunity', 'or',
             'original', 'orogastric', 'other', 'otherwise', 'overall',
             'overinflated', 'overlying', 'overt', 'pacemaker', 'parenchymal',
             'patchy', 'pathology', 'peribronchial', 'perihilar', 'persistent',
             'personally', 'personalname', 'phone', 'phones', 'physician',
             'picc', 'place', 'plain', 'please', 'pleural', 'pm', 'pneumonia',
             'pneumonitis', 'pneumothorax', 'portable', 'portions', 'position',
             'positions', 'possible', 'post', 'postoperative', 'postprocedure',
             'present', 'previous', 'printed', 'prior', 'probable', 'procedure',
             'procedures', 'process', 'prominence', 'prominent', 'proper',
             'provide', 'proximal', 'pulmonary', 'question', 'questionable',
             'rad', 'radiograph', 'radiographic', 'radiographically',
             'radiographs', 'radiography', 'radiologist', 'radiology', 'raise',
             'ray', 'rays', 'reactive', 'referral', 'reflect', 'region',
             'relate', 'relatively', 'remain', 'remains', 'removal', 'report',
             'reported', 'repositioning', 'represent', 'representing',
             'resident', 'respiratory', 'reviewed', 'right', 'satisfactory',
             'scarring', 'seen', 'segmental', 'senescent', 'setting',
             'settings', 'shallow', 'sided', 'sign', 'signature', 'signed',
             'signer', 'significant', 'signing', 'silhouette', 'similar',
             'single', 'size', 'slightly', 'small', 'soft', 'some', 'specific',
             'stable', 'standard', 'stat', 'status', 'sternotomy', 'stomach',
             'streaky', 'study', 'subsegmental', 'subtle', 'such', 'suggest',
             'suggesting', 'superimposed', 'superior', 'supervised', 'support',
             'suspicious', 'svc', 'swan', 'technologist', 'terminates', 'than',
             'thank', 'that', 'the', 'there', 'these', 'thickening', 'this',
             'thoracentesis', 'thoracic', 'throughout', 'time', 'tip', 'tissue',
             'to', 'trace', 'trachea', 'tract', 'transcribed',
             'transcriptionist', 'tube', 'tubes', 'two', 'unchanged',
             'unremarkable', 'upper', 'uva', 'vascular', 'vena', 'venous',
             'versus', 'view', 'views', 'viral', 'visible', 'visualized',
             'volumes', 'voluntary', 'vr', 'was', 'well', 'which', 'with',
             'within', 'without', 'wording', 'workstation', 'worse',
             'worsening', 'yesterday', 'you', 'your', 'zip'], dtype=object)
```

[24]:
```
X_findings
X_clinicaldata
X_examname
X_impression
```

[24]:
```
<10264x512 sparse matrix of type '<class 'numpy.float64'>'
        with 11057 stored elements in Compressed Sparse Row format>
```

[25]:
```
X_findings.toarray()
X_clinicaldata.toarray()
X_examname.toarray()
X_impression.toarray()
```

[25]:
```
array([[0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       …,
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.]])
```

[26]:
```
X_embeddedF = TSNE(n_components=2,␣
 ↪learning_rate='auto',init='random',perplexity=3).fit_transform(X_findings)
```

```
X_embeddedC = TSNE(n_components=2,␣
 ↪learning_rate='auto',init='random',perplexity=3).
 ↪fit_transform(X_clinicaldata)
X_embeddedE = TSNE(n_components=2,␣
 ↪learning_rate='auto',init='random',perplexity=3).fit_transform(X_examname)
X_embeddedI = TSNE(n_components=2,␣
 ↪learning_rate='auto',init='random',perplexity=3).fit_transform(X_impression)
```

[27]:

```
X_embeddedF
X_embeddedC
X_embeddedE
X_embeddedI
```

[27]:

```
array([[ -3.571836 ,  65.39446  ],
       [ 16.008808 ,   3.8225982],
       [ -4.525829 ,  -8.452661 ],
       …,
       [-76.60822  ,  37.996254 ],
       [-16.878683 , -94.87263  ],
       [-61.243355 ,  46.772808 ]], dtype=float32)
```

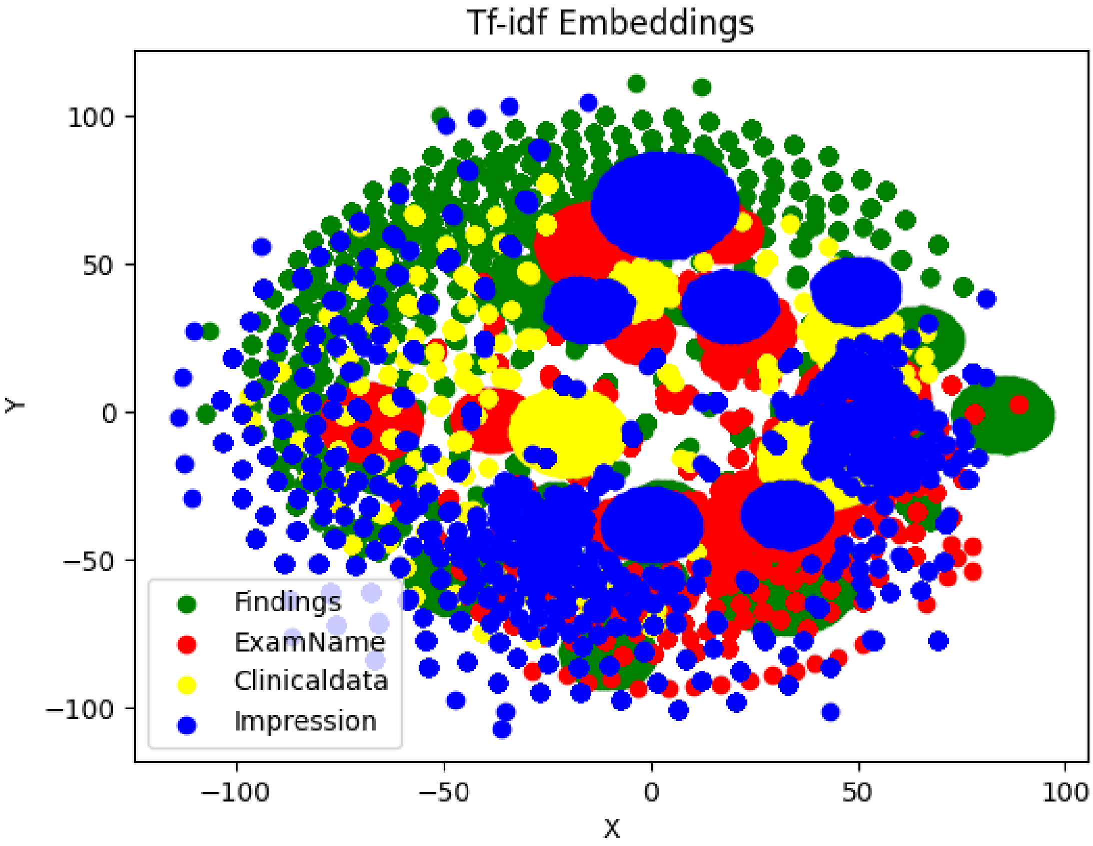
[32]:

```
plt.title("Tf-idf Embeddings")
plt.xlabel("X")
plt.ylabel("Y")

# Notation :,# takes all the columns from the number
plt.scatter(X_embeddedF[:,0], X_embeddedF[:,1], c='green')
plt.scatter(X_embeddedE[:,0], X_embeddedE[:,1], c='red')
plt.scatter(X_embeddedC[:,0], X_embeddedC[:,1], c='yellow')
plt.scatter(X_embeddedI[:,0], X_embeddedI[:,1], c='blue')
plt.legend(['Findings', 'ExamName','Clinicaldata','Impression'])
```

[32]:

```
<matplotlib.legend.Legend at 0x22460bc19c0>
```


Tf-idf Embeddings
100
50
0
−50
−100
Y
−100
−50
0
50
100
X
Findings
ExamName
Clinicaldata
Impression

[29]:
```
plt.show()
```