Baseline_model

September 17, 2020

```
[1]: import numpy as np
     import pandas as pd

     from baseline.torch.mydataset import CustomDataset
     from baseline.torch.model import CNN_Text
     from baseline.torch.train import train, predict_prob, run

     import torch
     from torchtext import data
     import time
     from tqdm import tqdm
```

1 load data

```
[2]: data_root = './data/'
```

```
[3]: drugs = ['trametinib',
              'fulvestrant',
              'lovastatin',
              'abiraterone',
              'thalidomide',
              'sirolimus',
              'simvastatin',
              'methotrexate',
              'bortezomib',
              'gemcitabine',
              'tamoxifen',
              'dexamethasone',
              'doxorubicin']
     len(drugs)
```

```
[3]: 13
```

```
[4]: alllab_df = pd.read_csv(data_root+'lab_finfin.csv')
     alllab_df.head()
```

```
[4]:         drug                                         file  lab comment
     0  tamoxifen  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3000794.xml    0    none
     1  tamoxifen  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3005955.xml    0    none
     2  tamoxifen  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3010527.xml    0    none
     3  tamoxifen  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3011858.xml    0    none
     4  tamoxifen  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3014261.xml    0    none
```

```
[11]: allfea_df = pd.read_csv(data_root+'fea_finfin.csv')
      allfea_df.head()
```

```
[11]:                                           file  \
      0  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3000794.xml
      1  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3001231.xml
      2  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3003872.xml
      3  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3004744.xml
      4  PMC003XXXXXX.xml\PMC0030XXXXX\PMC3005850.xml

                                                  title  \
      0  erk1 2 dependent vascular endothelial growth f…
      1  peg functionalized magnetic nanoparticles for …
      2  combination testing \( stage 2 \) of rapamycin…
      3  durable responses with the metronomic regimen …
      4  ph sensitive ionomeric particles obtained via …

                                               abstract
      0  background amp aims severe polycystic liver di…
      1  purpose polyethylene glycol ( peg ) functional…
      2  purpose rapamycin demonstrated broad spectrum …
      3  background targeting the tumor microenvironmen…
      4  silk fibroin based biomaterials have been wide…
```

```
[12]: train_lab_df = alllab_df[alllab_df['drug'].isin(drugs[:8])].
       ↪reset_index(drop=True)
      test_lab_df = alllab_df[alllab_df['drug'].isin(drugs[8:])].
       ↪reset_index(drop=True)
```

```
[13]: len_0 = len(train_lab_df[train_lab_df['lab']==0])
      len_1 = len(train_lab_df[train_lab_df['lab']==1])
      ratio = (len_0 - len_1)/len_1
```

```
[14]: train_lab_df_tmp = train_lab_df
      for _ in range(int(ratio)):
          train_lab_df_tmp = pd.concat([train_lab_df_tmp,␣
       ↪train_lab_df[train_lab_df['lab']==1]], ignore_index=True)
      train_lab_df =  train_lab_df_tmp
```

[15]:

```
len(train_lab_df[train_lab_df['lab']==0]),␣
 ↪len(train_lab_df[train_lab_df['lab']==1])
```

[15]:

```
(847, 828)
```

2 Build Vocabulary

[16]:

```
start_t = time.time()

text_field = data.Field(lower=True) # Text field
label_field = data.Field(sequential=False) # Label field

train_data, dev_data = CustomDataset.splits(text_field, label_field,␣
 ↪train_lab_df, allfea_df, shuffle=True)
test_data = CustomDataset(text_field, label_field, test_lab_df, allfea_df)

end_t = time.time()-start_t
print("Time elapse (min): ", end_t/60)
```

```
  4%|          | 69/1675 [00:00<00:02, 688.49it/s]

preparing examples…

100%|      | 1675/1675 [00:02<00:00, 567.31it/s]
  6%|          | 148/2681 [00:00<00:03, 741.43it/s]

dev_index:  -167
preparing examples…

100%|      | 2681/2681 [00:03<00:00, 729.41it/s]

Time elapse (min):  0.11108090082804362
```

[21]:

```
batch_size = 32
text_field.build_vocab(train_data, dev_data, test_data)
label_field.build_vocab(train_data, dev_data, test_data)
train_iter, dev_iter = data.Iterator.splits((train_data, dev_data),
                                            batch_sizes=(batch_size,␣
 ↪len(dev_data)))
```

3 Run the Baseline Model

[22]:

```
fields = [('text', text_field), ('label', label_field)]
```

```
[23]: model_dir_root = './trained_models/'
```

```
[24]: run(CNN_Text, model_dir_root+'model_baseline_fin.pkl', drugs[8:], alllab_df,␣
       ↪allfea_df, fields)
```

```
  6%|          | 24/371 [00:00<00:01, 234.32it/s]

########################
#### drug bortezomib

100%|      | 371/371 [00:01<00:00, 242.17it/s]
  7%|          | 28/415 [00:00<00:01, 272.90it/s]

The first paper is PMC004XXXXXX.xml\PMC0042XXXXX\PMC4266584.xml
Number of papers be read of drug [bortezomib]: 119
########################
#### drug gemcitabine

100%|      | 415/415 [00:01<00:00, 254.81it/s]
  5%|          | 28/526 [00:00<00:01, 279.68it/s]

The first paper is PMC004XXXXXX.xml\PMC0048XXXXX\PMC4873426.xml
Number of papers be read of drug [gemcitabine]: 5
########################
#### drug tamoxifen

100%|      | 526/526 [00:01<00:00, 263.32it/s]
  5%|          | 28/565 [00:00<00:01, 278.02it/s]

The first paper is PMC003XXXXXX.xml\PMC0037XXXXX\PMC3711713.xml
Number of papers be read of drug [tamoxifen]: 46
########################
#### drug dexamethasone

100%|      | 565/565 [00:02<00:00, 267.28it/s]
  4%|          | 29/804 [00:00<00:02, 282.54it/s]

The first paper is PMC004XXXXXX.xml\PMC0044XXXXX\PMC4422178.xml
Number of papers be read of drug [dexamethasone]: 141
########################
#### drug doxorubicin

100%|      | 804/804 [00:02<00:00, 271.49it/s]

The first paper is PMC003XXXXXX.xml\PMC0032XXXXX\PMC3298037.xml
Number of papers be read of drug [doxorubicin]: 384
```

```
[24]: ({'bortezomib': 119,
        'gemcitabine': 5,
        'tamoxifen': 46,
        'dexamethasone': 141,
        'doxorubicin': 384},
       {'bortezomib': 'PMC004XXXXXX.xml\\PMC0042XXXXX\\PMC4266584.xml',
        'gemcitabine': 'PMC004XXXXXX.xml\\PMC0048XXXXX\\PMC4873426.xml',
        'tamoxifen': 'PMC003XXXXXX.xml\\PMC0037XXXXX\\PMC3711713.xml',
        'dexamethasone': 'PMC004XXXXXX.xml\\PMC0044XXXXX\\PMC4422178.xml',
        'doxorubicin': 'PMC003XXXXXX.xml\\PMC0032XXXXX\\PMC3298037.xml'})
```