create_yelp_review_data

September 29, 2021

1 Create Yelp Reviews data for Sentiment Analysis and Word Embeddings

1.1 Imports & Settings

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
from pathlib import Path
import pandas as pd
from pandas.io.json import json_normalize
```

1.2 About the Data

The data consists of several files with information on the business, the user, the review and other aspects that Yelp provides to encourage data science innovation.

The data consists of several files with information on the business, the user, the review and other aspects that Yelp provides to encourage data science innovation.

We will use around six million reviews produced over the 2010-2019 period to extract text features. In addition, we will use other information submitted with the review about the user.

1.3 Getting the Data

You can download the data from here in json format after accepting the license. The 2020 version has 4.7GB (compressed) and around 10.5GB (uncompressed) of text data.

After download, extract the following two of the five `.json` files into to `./yelp/json`: - the `yelp_academic_dataset_user.json` - the `yelp_academic_dataset_reviews.json`

Rename both files by stripping out the `yelp_academic_dataset_` prefix so you have the following directory structure:

```
data
|-create_yelp_review_data.ipynb
|-yelp
    |-json
        |-user.json
        |-review.json
```

[3]:

```
yelp_dir = Path('yelp')

if not yelp_dir.exists():
    yelp_dir.mkdir(exist_ok=True)
```

1.4 Parse json and store as parquet files

Convert json to faster parquet format:

[4]:

```
for fname in ['review', 'user']:
    print(fname)

    json_file = yelp_dir / 'json' / f'{fname}.json'
    parquet_file = yelp_dir / f'{fname}.parquet'
    if parquet_file.exists():
        print('\talready exists')
        continue

    data = json_file.read_text(encoding='utf-8')
    json_data = '[' + ','.join([l.strip()
                                for l in data.split('\n') if l.strip()]) + ']\n'
    data = json.loads(json_data)
    df = json_normalize(data)
    if fname == 'review':
        df.date = pd.to_datetime(df.date)
        latest = df.date.max()
        df['year'] = df.date.dt.year
        df['month'] = df.date.dt.month
        df = df.drop(['date', 'business_id', 'review_id'], axis=1)
    if fname == 'user':
        df.yelping_since = pd.to_datetime(df.yelping_since)
        df = (df.assign(member_yrs=lambda x: (latest - x.yelping_since)
                        .dt.days.div(365).astype(int))
              .drop(['elite', 'friends', 'name', 'yelping_since'], axis=1))
    df.dropna(how='all', axis=1).to_parquet(parquet_file)
```

```
review
user
```

Now you can remove the json files.

[8]:

```
def merge_files(remove=False):
    combined_file = yelp_dir / 'user_reviews.parquet'
    if not combined_file.exists():
        user = pd.read_parquet(yelp_dir / 'user.parquet')
        print(user.info(null_counts=True))

        review = pd.read_parquet(yelp_dir / 'review.parquet')
        print(review.info(null_counts=True))

        combined = (review.merge(user, on='user_id',
                                 how='left', suffixes=['', '_user'])
                    .drop('user_id', axis=1))
        combined = combined[combined.stars > 0]
        print(combined.info(null_counts=True))
        combined.to_parquet(yelp_dir / 'user_reviews.parquet')
    else:
        print('already merged')
    if remove:
        for fname in ['user', 'review']:
            f = yelp_dir / (fname + '.parquet')
            if f.exists():
                f.unlink()
```

```
[9]: merge_files(remove=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1968703 entries, 0 to 1968702
Data columns (total 19 columns):
 #   Column              Non-Null Count    Dtype
---  ------              --------------    -----
 0   user_id             1968703 non-null  object
 1   review_count        1968703 non-null  int64
 2   useful              1968703 non-null  int64
 3   funny               1968703 non-null  int64
 4   cool                1968703 non-null  int64
 5   fans                1968703 non-null  int64
 6   average_stars       1968703 non-null  float64
 7   compliment_hot      1968703 non-null  int64
 8   compliment_more     1968703 non-null  int64
 9   compliment_profile  1968703 non-null  int64
 10  compliment_cute     1968703 non-null  int64
 11  compliment_list     1968703 non-null  int64
 12  compliment_note     1968703 non-null  int64
 13  compliment_plain    1968703 non-null  int64
 14  compliment_cool     1968703 non-null  int64
 15  compliment_funny    1968703 non-null  int64
 16  compliment_writer   1968703 non-null  int64
 17  compliment_photos   1968703 non-null  int64
 18  member_yrs          1968703 non-null  int64
dtypes: float64(1), int64(17), object(1)
memory usage: 285.4+ MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8021122 entries, 0 to 8021121
```

```
Data columns (total 8 columns):
 #   Column   Non-Null Count    Dtype  
---  ------   --------------    -----  
 0   user_id  8021122 non-null  object 
 1   stars    8021122 non-null  float64
 2   useful   8021122 non-null  int64  
 3   funny    8021122 non-null  int64  
 4   cool     8021122 non-null  int64  
 5   text     8021122 non-null  object 
 6   year     8021122 non-null  int64  
 7   month    8021122 non-null  int64  
dtypes: float64(1), int64(5), object(2)
memory usage: 489.6+ MB
None
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8021122 entries, 0 to 8021121
Data columns (total 25 columns):
 #   Column              Non-Null Count    Dtype  
---  ------              --------------    -----  
 0   stars               8021122 non-null  float64
 1   useful              8021122 non-null  int64  
 2   funny               8021122 non-null  int64  
 3   cool                8021122 non-null  int64  
 4   text                8021122 non-null  object 
 5   year                8021122 non-null  int64  
 6   month               8021122 non-null  int64  
 7   review_count        8021122 non-null  int64  
 8   useful_user         8021122 non-null  int64  
 9   funny_user          8021122 non-null  int64  
 10  cool_user           8021122 non-null  int64  
 11  fans                8021122 non-null  int64  
 12  average_stars       8021122 non-null  float64
 13  compliment_hot      8021122 non-null  int64  
 14  compliment_more     8021122 non-null  int64  
 15  compliment_profile  8021122 non-null  int64  
 16  compliment_cute     8021122 non-null  int64  
 17  compliment_list     8021122 non-null  int64  
 18  compliment_note     8021122 non-null  int64  
 19  compliment_plain    8021122 non-null  int64  
 20  compliment_cool     8021122 non-null  int64  
 21  compliment_funny    8021122 non-null  int64  
 22  compliment_writer   8021122 non-null  int64  
 23  compliment_photos   8021122 non-null  int64  
 24  member_yrs          8021122 non-null  int64  
dtypes: float64(2), int64(22), object(1)
memory usage: 1.6+ GB
None
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-9-892254c45114> in <module>
----> 1 merge_files(remove=True)

<ipython-input-8-159c5a2caa96> in merge_files(remove)
     18     if remove:
     19         for fname in ['user', 'reviews']:
---> 20             f = yelp_dir / fname + '.parquet'
     21             if f.exists():
     22                 f.unlink()

TypeError: unsupported operand type(s) for +: 'PosixPath' and 'str'
```