02_ceeaus_clean

March 21, 2017

2/24/17 - smiel

1 Cleaning the CEEAUS essay data.

```
In [2]: %matplotlib inline
        # run me when first starting this notebook
        import os

        import numpy as np
        import pandas as pd

        path = '/research/ella/rivendell/ceeaus'
```

The CEEAUS data comes to us in a bunch of text files, in L1 folders. We'll use these to create an essays csv.

```
In [6]: import codecs
        recs = []
        l1_map = {'chinese': 'CHN', 'english': 'ENG', 'japanese': 'JPN'}
        toeic_map = {'L': 250.0, 'M': 550.0, 'S': 650.0, 'U': 850.0}
        data_path = os.path.join(path, 'ceeaus')

        for lang, l1 in l1_map.items():
            lang_path = os.path.join(data_path, lang)
            for filename in os.listdir(lang_path):
                rec = {'dataset': 'CEEAUS', 'essay_id': '{}/{}'.format(lang, filename)}
                level = None
                if l1 == 'JPN':
                    _, level, prompt_id, _ = filename[:-4].split('_')
                else:
                    _, prompt_id, _ = filename[:-4].split('_')

                rec['prompt_id'] = prompt_id
                rec['L1'] = l1
                rec['student_level_TOEIC'] = np.nan if level is None else toeic_map[level]

                with codecs.open(os.path.join(lang_path, filename), 'r', encoding='shift_jis_200
```

```
                rec['text'] = fin.read().strip()

            recs.append(rec)

    df = pd.DataFrame.from_records(recs)
    df.to_csv(os.path.join(path, 'all_essays.csv'), encoding='shift_jis_2004', index=False)
    df.head()
```

Out[6]:

	L1	dataset	essay_id	prompt_id	student_level_TOEIC	\
0	JPN	CEEAUS	japanese/ceejus_L_smk_32.txt	smk	250.0	
1	JPN	CEEAUS	japanese/ceejus_M_ptj_108.txt	ptj	550.0	
2	JPN	CEEAUS	japanese/ceejus_S_smk_083.txt	smk	650.0	
3	JPN	CEEAUS	japanese/ceejus_M_ptj_133.txt	ptj	550.0	
4	JPN	CEEAUS	japanese/ceejus_S_smk_051.txt	smk	650.0	

	text
0	Tabacco contains so many sorts of poison. For ...
1	"I agree this statement. This is because an ex...
2	I think that all restaurants should forbid the...
3	I think it is important for college students t...
4	We should permit smokers to smoke at the resta...

```
In [5]: df.text.values[0]
```

```
Out[5]: u'Tabacco contains so many sorts of poison. For instance, nikotin is most famous poison
```

1.1 From Essays to Sentences

Now let's start building the sentences data frame. For unidecode to work properly, the following should print "True":

```
In [11]: import sys
         print(sys.maxunicode > 0xffff)
```

```
True
```

```
In [3]: from utilitybelt.text import get_sentences
        import copy
        from unidecode import unidecode
        import numpy as np

        # load data
        df_in = pd.read_csv(os.path.join(path, 'all_essays.csv'), encoding='shift_jis_2004')

        # convert text to ascii
        print('Converting to ASCII')
        df_in['ascii_text'] = df_in.text.apply(lambda t: unidecode(t))
```

```
# normalize line endings
df_in.ascii_text = df_in.ascii_text.str.replace('\r\n', '\n')
df_in.ascii_text = df_in.ascii_text.str.replace('\r', '\n')

# use space instead of tab
df_in.ascii_text = df_in.ascii_text.str.replace('\t', ' ')

# now remove any non-printable ascii char
df_in.ascii_text = df_in.ascii_text.str.replace(r'[^ -~\n]', '')

# # make sure all is printable
# for i, t in enumerate(df_in.ascii_text.values):
#     for ci, c in enumerate(t):
#         if (32 <= ord(c) <= 126) or c in '\n\t':
#             continue
#         else:
#             print u"Unprintable character {} in {} at char {}:\n\n{}\n================
#                 ord(c), i, ci, t, df_in.iloc[i].clean_text
#             )
#             raise ValueError

# shush the utilitybelt sentence splitter logging
import logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)

def asfloat(x):
    try:
        return float(x)
    except:
        print '{} not a float'.format(x)
        return np.nan

print('Splitting sentences')
# create records for every sentence
records = []
for i, row in df_in.iterrows():
    rec = {
        'dataset': row.dataset, 'prompt_id': row.prompt_id, 'essay_id': row.essay_id,
        'L1': row.L1, 'score': np.nan, 'score_type': '', 'age': np.nan,
        'student_level_TOEIC': row.student_level_TOEIC,
    }
    prev_end = 0
    text = row.ascii_text
    si = 0
    for start, end, sentence in zip(*get_sentences(text)):
        srec = {}
        srec.update(rec)
```

```
                srec['text'] = sentence
                srec['sentence_id'] = si
                srec['trailing_whitespace'] = text[prev_end:start]
                si += 1
                prev_end = end
                records.append(srec)

            if i % 1000 == 0:
                print('{} of {}'.format(i, len(df_in)))

        print('Creating data frame')
        df_out = pd.DataFrame.from_records(records)
        df_out['uid'] = df_out[['dataset', 'essay_id', 'sentence_id']].astype(unicode).apply(lam

        print('{} sentences'.format(len(df_out)))
        print('Saving data frame')
        df_out.to_csv(os.path.join(path, 'CEEAUS_sentences.csv'), encoding='utf8', index=False)
```

```
Converting to ASCII
Splitting sentences
0 of 1008
1000 of 1008
Creating data frame
15758 sentences
Saving data frame
```

Let's do a little descriptive analysis to make sure we got what we want.

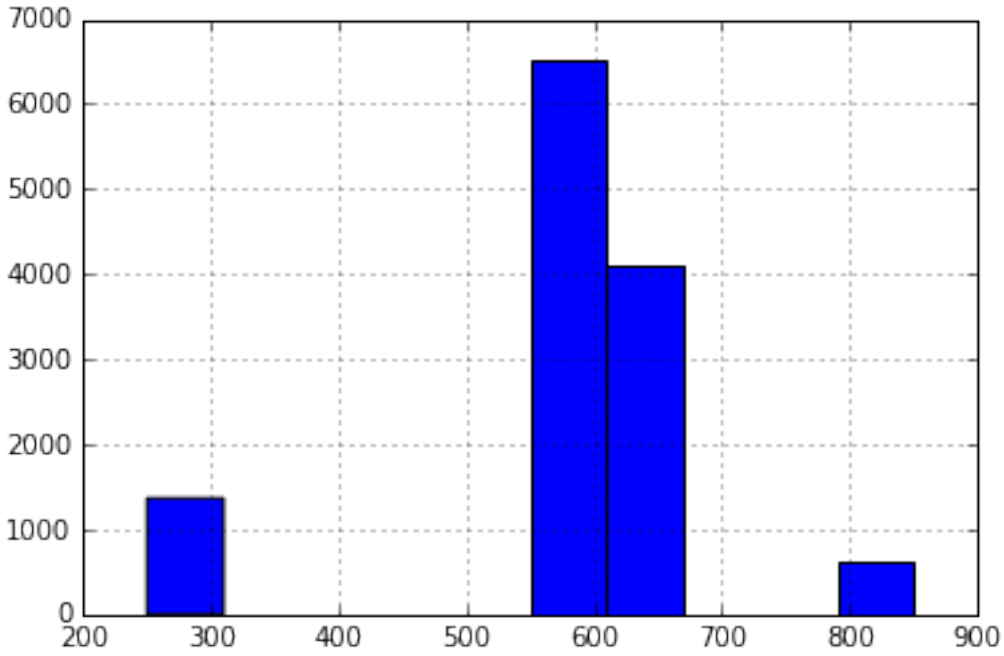
```
In [4]: df = pd.read_csv(os.path.join(path, 'CEEAUS_sentences.csv'), encoding='utf8')

In [5]: len(df.groupby('essay_id'))

Out[5]: 1008

In [10]: df.student_level_TOEIC[pd.notnull(df.student_level_TOEIC)].hist()

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f31176e5810>
```




```
In [11]: age = df.groupby('age').size()
         print(age)
         print('{} sentences with age data'.format(pd.notnull(df.age).sum()))

Series([], dtype: int64)
0 sentences with age data


In [12]: score = df.groupby('score').size()
         print(score)

Series([], dtype: int64)


In [13]: df.text.apply(len).describe()

Out[13]: count    15758.000000
         mean        77.976520
         std         42.143806
         min          1.000000
         25%         49.000000
         50%         71.000000
         75%         97.000000
         max        455.000000
         Name: text, dtype: float64
```

```
In [14]: df.trailing_whitespace = df.trailing_whitespace.fillna('')
         essay1_id = df.essay_id.values[0]
         essay1 = df[df.essay_id == essay1_id]
         essay1['text_plus'] = essay1.trailing_whitespace + essay1.text
         text = ''.join(essay1.text_plus.values)
         print(text)
         print(essay1_id)
```

```
Tabacco contains so many sorts of poison. For instance, nikotin is most famous poison of them. S
japanese/ceejus_L_smk_32.txt
```

```
/home/smiel/.venvs/rivendell/lib/python2.7/site-packages/ipykernel/__main__.py:4: SettingWithCop
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
```