train_dev_test

March 21, 2017

02/27/17 - smiel ## Generating train, dev and test data sets from all_essays.csv

In [2]:
```
import numpy as np
import pandas as pd
import os

path = '/research/ella/rivendell'
df = pd.read_csv(os.path.join(path, 'all_essays.csv'), encoding='utf8')
```

```
/home/smiel/.venvs/rivendell/local/lib/python2.7/site-packages/IPython/core/interactiveshell.py:
  interactivity=interactivity, compiler=compiler, result=result)
```

In [3]:
```
# How many english essays are in the test set?
len(df[df.dataset.isin(['ICNALE', 'CEEAUS', 'MOECS']) & (df.L1 == 'ENG')])
```

Out[3]: 591

In [4]:
```
# How many non-english essays are in the test set?
len(df[df.dataset.isin(['ICNALE', 'CEEAUS', 'MOECS']) & (df.L1 != 'ENG')])
```

Out[4]: 5549

In [5]:
```
# How many essays are in FCE?
(df.dataset == 'FCE').sum()
```

Out[5]: 2481

We'll keep the dev set balanced, so let's add English essays (from ASAP) to FCE to balance the classes.

In [6]:
```
import re

def replace_countries(frame):
    # precalculate prompt words and country names
    foreign_countries = [
        'Afghanistan',
        'Albania',
        'Algeria',
```

```
'Andorra',
'Angola',
'Antigua',
'Barbuda',
'Argentina',
'Armenia',
'Aruba',
'Australia',
'Austria',
'Azerbaijan',
'Bahamas',
'Bahrain',
'Bangladesh',
'Barbados',
'Belarus',
'Belgium',
'Belize',
'Benin',
'Bhutan',
'Bolivia',
'Bosnia',
'Herzegovina',
'Botswana',
'Brazil',
'Brunei',
'Bulgaria',
'Burkina Faso',
'Burma',
'Burundi',
'Cambodia',
'Cameroon',
'Canada',
'Cabo Verde',
'Central African Republic',
'Chad',
'Chile',
'China',
'Colombia',
'Comoros',
'(Democratic )?(Republic of (the )?)?Congo',
'Costa Rica',
'Cote d\'Ivoire',
'Croatia',
'Cuba',
'Curacao',
'Cyprus',
'Czechia',
'Denmark',
```

```
'Djibouti',
'Dominica',
'Dominican Republic',
'Ecuador',
'Egypt',
'El Salvador',
'Equatorial Guinea',
'Eritrea',
'Estonia',
'Ethiopia',
'Fiji',
'Finland',
'France',
'Gabon',
'Gambia',
'Georgia',
'Germany',
'Ghana',
'Greece',
'Grenada',
'Guatemala',
'Guinea',
'Guinea-Bissau',
'Guyana',
'Haiti',
'Holy See',
'Honduras',
'Hong Kong',
'Hungary',
'Iceland',
'India',
'Indonesia',
'Iran',
'Iraq',
'Ireland',
'Israel',
'Italy',
'Jamaica',
'Japan',
'Jordan',
'Kazakhstan',
'Kenya',
'Kiribati',
'((North|South) )?Korea',
'Kosovo',
'Kuwait',
'Kyrgyzstan',
'Laos',
```

```
'Latvia',
'Lebanon',
'Lesotho',
'Liberia',
'Libya',
'Liechtenstein',
'Lithuania',
'Luxembourg',
'Macau',
'Macedonia',
'Madagascar',
'Malawi',
'Malaysia',
'Maldives',
'Mali',
'Malta',
'Marshall Islands',
'Mauritania',
'Mauritius',
'Mexico',
'Micronesia',
'Moldova',
'Monaco',
'Mongolia',
'Montenegro',
'Morocco',
'Mozambique',
'Namibia',
'Nauru',
'Nepal',
'Netherlands',
'New Zealand',
'Nicaragua',
'Niger',
'Nigeria',
'North Korea',
'Norway',
'Oman',
'Pakistan',
'Palau',
'Palestinian Territories',
'Palestine',
'Panama',
'Papua New Guinea',
'Paraguay',
'Peru',
'Philippines',
'Poland',
```

```
'Portugal',
'Qatar',
'Romania',
'Russia',
'Rwanda',
'Saint Kitts and Nevis',
'Saint Lucia',
'Saint Vincent and the Grenadines',
'Samoa',
'San Marino',
'Sao Tome and Principe',
'Saudi Arabia',
'Senegal',
'Serbia',
'Seychelles',
'Sierra Leone',
'Singapore',
'Sint Maarten',
'Slovakia',
'Slovenia',
'Solomon Islands',
'Somalia',
'South Africa',
'South Korea',
'South Sudan',
'Spain',
'Sri Lanka',
'Sudan',
'Suriname',
'Swaziland',
'Sweden',
'Switzerland',
'Syria',
'Taiwan',
'Tajikistan',
'Tanzania',
'Thailand',
'Timor-Leste',
'Togo',
'Tonga',
'Trinidad and Tobago',
'Tunisia',
'Turkey',
'Turkmenistan',
'Tuvalu',
'Uganda',
'Ukraine',
'United Arab Emirates',
```

```
            'United Kingdom',
            'Uruguay',
            'Uzbekistan',
            'Vanuatu',
            'Venezuela',
            'Vietnam',
            'Yemen',
            'Zambia',
            'Zimbabwe',
        ]
        usa_names = [
            '(United States of )?America',
            'United States',
            'U\.?S\.?A\.?',
        ]

        usa_regex = '|'.join(['({})'.format(c) for c in usa_names])

        for c in foreign_countries:
            print(c)
            frame.text = frame.text.str.replace(c, 'COUNTRY', flags=re.IGNORECASE)

        print('usa')
        frame.text = frame.text.str.replace(usa_regex, 'COUNTRY', flags=re.IGNORECASE)

        return frame
```

In [7]:
```
df['non_native'] = df.L1 != 'ENG'

# drop too short or too long
df = df[(df.text.str.len() >= 150) & (df.text.str.len() <= 4000)]

# ICNALE, CEEAUS, and MOECS
test = df[df.dataset.isin(['ICNALE', 'CEEAUS', 'MOECS'])].copy()
df = df[~(df.dataset.isin(['ICNALE', 'CEEAUS', 'MOECS']))]

# replace country names
df = replace_countries(df)

# FCE and a subsample of ASAP
asap_sub = df[df.dataset == 'ASAP'].sample(n=2481, random_state=42)
dev = df[(df.dataset == 'FCE') | (df.index.isin(asap_sub.index))]

# everything else from the other data sets
train = df[~(df.dataset.isin(['ICNALE', 'CEEAUS', 'MOECS', 'FCE', 'ASAP']))]

# turn prompts into ints
train['dataset_prompt'] = train.apply(lambda row: '{}_{}'.format(row.dataset, row.prompt
```

```
        prompt_map = {}
        for dp in set(train.dataset_prompt):
            prompt_map[dp] = len(prompt_map)

        train['prompt_index'] = train.dataset_prompt.map(prompt_map)
        max_prompt = train.prompt_index.max() + 1

        # turn prompts into ints
        dev['dataset_prompt'] = dev.apply(lambda row: '{}_{}'.format(row.dataset, row.prompt_id)
        prompt_map = {}
        for dp in set(dev.dataset_prompt):
            prompt_map[dp] = len(prompt_map) + max_prompt

        dev['prompt_index'] = dev.dataset_prompt.map(prompt_map)
        max_prompt = dev.prompt_index.max() + 1

        # turn prompts into ints
        test['dataset_prompt'] = test.apply(lambda row: '{}_{}'.format(row.dataset, row.prompt_i
        prompt_map = {}
        for dp in set(test.dataset_prompt):
            prompt_map[dp] = len(prompt_map) + max_prompt

        test['prompt_index'] = test.dataset_prompt.map(prompt_map)

        print(sorted(set(train.prompt_index)))
        print(sorted(set(dev.prompt_index)))
        print(sorted(set(test.prompt_index)))
```

```
/home/smiel/.venvs/rivendell/lib/python2.7/site-packages/ipykernel/__main__.py:11: UserWarning:


Afghanistan
Albania
Algeria
Andorra
Angola
Antigua
Barbuda
Argentina
Armenia
Aruba
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
```

```
Belarus
Belgium
Belize
Benin
Bhutan
Bolivia
Bosnia
Herzegovina
Botswana
Brazil
Brunei
Bulgaria
Burkina Faso
Burma
Burundi
Cambodia
Cameroon
Canada
Cabo Verde
Central African Republic
Chad
Chile
China
Colombia
Comoros
(Democratic )?(Republic of (the )?)?Congo
Costa Rica
Cote d'Ivoire
Croatia
Cuba
Curacao
Cyprus
Czechia
Denmark
Djibouti
Dominica
Dominican Republic
Ecuador
Egypt
El Salvador
Equatorial Guinea
Eritrea
Estonia
Ethiopia
Fiji
Finland
France
Gabon
```

Gambia
Georgia
Germany
Ghana
Greece
Grenada
Guatemala
Guinea
Guinea-Bissau
Guyana
Haiti
Holy See
Honduras
Hong Kong
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Jamaica
Japan
Jordan
Kazakhstan
Kenya
Kiribati
((North|South))?Korea
Kosovo
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Luxembourg
Macau
Macedonia
Madagascar
Malawi
Malaysia
Maldives

Mali
Malta
Marshall Islands
Mauritania
Mauritius
Mexico
Micronesia
Moldova
Monaco
Mongolia
Montenegro
Morocco
Mozambique
Namibia
Nauru
Nepal
Netherlands
New Zealand
Nicaragua
Niger
Nigeria
North Korea
Norway
Oman
Pakistan
Palau
Palestinian Territories
Palestine
Panama
Papua New Guinea
Paraguay
Peru
Philippines
Poland
Portugal
Qatar
Romania
Russia
Rwanda
Saint Kitts and Nevis
Saint Lucia
Saint Vincent and the Grenadines
Samoa
San Marino
Sao Tome and Principe
Saudi Arabia
Senegal
Serbia

```
Seychelles
Sierra Leone
Singapore
Sint Maarten
Slovakia
Slovenia
Solomon Islands
Somalia
South Africa
South Korea
South Sudan
Spain
Sri Lanka
Sudan
Suriname
Swaziland
Sweden
Switzerland
Syria
Taiwan
Tajikistan
Tanzania
Thailand
Timor-Leste
Togo
Tonga
Trinidad and Tobago
Tunisia
Turkey
Turkmenistan
Tuvalu
Uganda
Ukraine
United Arab Emirates
United Kingdom
Uruguay
Uzbekistan
Vanuatu
Venezuela
Vietnam
Yemen
Zambia
Zimbabwe
usa

/home/smiel/.venvs/rivendell/lib/python2.7/site-packages/ipykernel/__main__.py:24: SettingWithCo
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#
/home/smiel/.venvs/rivendell/lib/python2.7/site-packages/ipykernel/__main__.py:29: SettingWithCo
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#
/home/smiel/.venvs/rivendell/lib/python2.7/site-packages/ipykernel/__main__.py:33: SettingWithCo
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#
/home/smiel/.venvs/rivendell/lib/python2.7/site-packages/ipykernel/__main__.py:38: SettingWithCo
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#
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 2
[95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114,
[140, 141, 142, 143, 144]
```

In [8]:
```
# cleanup for minimal loading issues. Can always retrieve originals from uid or index
nativeness_path = '/research/ella/nativeness'

# train
train = train[['uid', 'non_native', 'prompt_index', 'text']]
train.to_csv(os.path.join(nativeness_path, 'train.csv'), encoding='utf8')

# dev
dev = dev[['uid', 'non_native', 'prompt_index', 'text']]
dev.to_csv(os.path.join(nativeness_path, 'dev.csv'), encoding='utf8')

# test
test = test[['uid', 'non_native', 'prompt_index', 'text']]
test.to_csv(os.path.join(nativeness_path, 'test.csv'), encoding='utf8')
```

In [9]:
```
(df.text.str.len() < 150).sum()
```

Out[9]: 0

In [10]:
```
print('{}, {}, {}'.format(len(train), len(dev), len(test)))
```

```
66677, 4236, 6133
```