M05 Homework

Michael Vaden, mtv2eva

In [ ]:
```
import pandas as pd
import numpy as np
from glob import glob
import re
import nltk
import plotly_express as px
from lib.textparser import TextParser
```

In [ ]:
```
import configparser
config = configparser.ConfigParser()
config.read("../env.ini")
data_home = config['DEFAULT']['data_home']
output_dir = config['DEFAULT']['output_dir']
data_prefix = 'austen-melville'
```

In [ ]:
```
OHCO = ['book_id', 'chap_id', 'para_num', 'sent_num', 'token_num']
bags = dict(
    SENTS = OHCO[:4],
    PARAS = OHCO[:3],
    CHAPS = OHCO[:2],
    BOOKS = OHCO[:1]
)
```

In [ ]:
```
LIB = pd.read_csv(f"{output_dir}/{data_prefix}-LIB.csv").set_index('book_id')
CORPUS = pd.read_csv(f"{output_dir}/{data_prefix}-CORPUS.csv").set_index(OHCO)
VOCAB = pd.read_csv(f'{output_dir}/{data_prefix}-VOCAB.csv').set_index('term_str').dropna()
```

Question 1: Show the function you created.

In [ ]:
```
def create_bag_of_words(CORPUS, bag):
    BOW = CORPUS.groupby(bag+['term_str']).term_str.count().to_frame('n')
    return BOW
```

In [ ]:
```
idf_method = 'standard'
```

In [ ]:
```
def get_TFIDF(BOW, tf_method):
    DTCM = BOW.n.unstack(fill_value=0)

    DF = DTCM.astype('bool').sum()
    N = len(DTCM)

    if tf_method == 'sum':
        TF = DTCM.T / DTCM.T.sum()

    elif tf_method == 'max':
        TF = DTCM.T / DTCM.T.max()

    elif tf_method == 'log':
        TF = np.log2(1 + DTCM.T)

    elif tf_method == 'raw':
        TF = DTCM.T

    elif tf_method == 'double_norm':
        TF = DTCM.T / DTCM.T.max()

    elif tf_method == 'binary':
        TF = DTCM.T.astype('bool').astype('int')

    TF = TF.T

    if idf_method == 'standard':
        IDF = np.log2(N / DF)

    elif idf_method == 'max':
        IDF = np.log2(DF.max() / DF)

    elif idf_method == 'smooth':
        IDF = np.log2((1 + N) / (1 + DF)) + 1

    return TF * IDF
```

Question 2: What are the top 20 words in the corpus by TFIDF mean using the `max` count method and `book` as the bag?

In [ ]:
```
CORPUS
```

Out[ ]:

book_id	chap_id	para_num	sent_num	token_num	pos_tuple	pos	token_str	term_str	pos_group
105	1	1	0	0	('Sir', 'NNP')	NNP	Sir	sir	NN
				1	('Walter', 'NNP')	NNP	Walter	walter	NN
				2	('Elliot,', 'NNP')	NNP	Elliot,	elliot	NN
				3	('of', 'IN')	IN	of	of	IN
				4	('Kellynch', 'NNP')	NNP	Kellynch	kellynch	NN
...	...	...	...	...	...	...	...	...	...
34970	114	24	0	6	('The', 'DT')	DT	The	the	DT
				7	('Ambiguities,', 'NNP')	NNP	Ambiguities,	ambiguities	NN
				8	('by', 'IN')	IN	by	by	IN
				9	('Herman', 'NNP')	NNP	Herman	herman	NN
				10	('Melville', 'NNP')	NNP	Melville	melville	NN

2059272 rows × 5 columns

In [ ]:
```
get_TFIDF(create_bag_of_words(CORPUS, bags['BOOKS']), 'max').mean().reset_index()\
    .rename({0:'AVG_TFIDF'}, axis=1).sort_values('AVG_TFIDF', ascending=False).head(20).reset_index(drop=True)
```

Out[ ]:

	term_str	AVG_TFIDF
0	elinor	0.033840
1	pierre	0.030911
2	vernon	0.025980
3	marianne	0.021347
4	emma	0.021164
5	darcy	0.019302
6	reginald	0.018486
7	babbalanja	0.018252
8	catherine	0.018238
9	frederica	0.017986
10	crawford	0.017749
11	fanny	0.017167
12	elliot	0.017053
13	weston	0.016591
14	media	0.015986
15	israel	0.015428
16	knightley	0.015184
17	tilney	0.013815
18	elton	0.013648
19	bingley	0.013264

Question 3: What are the top 20 words in the corpus by TFIDF mean, if you using the `sum` count method and `chapter` as the bag? Note, because of the greater number of bags, this will take longer to compute.

In [ ]:
```
get_TFIDF(create_bag_of_words(CORPUS, bags['CHAPS']), 'sum').mean().reset_index()\
    .rename({0:'AVG_TFIDF'}, axis=1).sort_values('AVG_TFIDF', ascending=False).head(20).reset_index(drop=True)
```

Out[ ]:

	term_str	AVG_TFIDF
0	her	0.004327
1	she	0.004150
2	cosmopolitan	0.003485
3	pierre	0.003317
4	communion	0.003004
5	i	0.002771
6	sailors	0.002668
7	you	0.002620
8	hypothetical	0.002437
9	mr	0.002084
10	and	0.002054
11	confidential	0.002042
12	the	0.001972
13	dream	0.001942
14	boon	0.001857
15	mrs	0.001747
16	elephants	0.001731
17	whale	0.001715
18	thou	0.001696
19	acquaintance	0.001690

Question 4: Characterize the general difference between the words in Question 3 and those in Question 2 in terms of part-of-speech.

The words in question 3 are all proper nouns, as compared to by chapter which is a combination of different parts of speech

Question 5: Compute mean `TFIDF` for vocabularies conditioned on individual author, using *chapter* as the bag and `max` as the `TF` count method. Among the two authors, whose work has the most significant adjective?

In [ ]:
```
BOW_JA = create_bag_of_words(CORPUS.reset_index().join(LIB.author, on='book_id').query("author == 'AUSTEN, JANE'"), bags['CHAPS'])
BOW_HM = create_bag_of_words(CORPUS.reset_index().join(LIB.author, on='book_id').query("author == 'MELVILLE, HERMAN'"), bags['CHAPS'])
```

In [ ]:
```
get_TFIDF(BOW_JA, 'max').mean().to_frame('AVG_TFIDF').join(VOCAB.max_pos).query("max_pos == 'JJ'")\
    .sort_values('AVG_TFIDF', ascending=False).head(5)
```

Out[ ]:

term_str	AVG_TFIDF	max_pos
sure	0.013167	JJ
dear	0.012992	JJ
poor	0.012213	JJ
upper	0.011347	JJ
old	0.011327	JJ

In [ ]:
```
get_TFIDF(BOW_HM, 'max').mean().to_frame('AVG_TFIDF').join(VOCAB.max_pos).query("max_pos == 'JJ'")\
    .sort_values('AVG_TFIDF', ascending=False).head(5)
```

Out[ ]:

term_str	AVG_TFIDF	max_pos
thy	0.028653	JJ
old	0.021042	JJ
ugh	0.015733	JJ
little	0.014585	JJ
good	0.014173	JJ

The most significant adjective is *thy* and belongs to Herman Melville