HW 2

Course: DS 5001
Module: 02 Text Models
Topic: HW 2
Author: Lindley Slipetz
Date: 24 January 2025

Set Up

Import libraries

In [1]:
```
import pandas as pd
```

Import Config

In [2]:
```
text_file = "C:\\Users\\ddj6tu\\Documents\\GitHub\\DS5001\\data\\pg161.txt"
csv_file  = "C:\\Users\\ddj6tu\\Documents\\GitHub\\DS5001\\data\\austen-sense.csv" # T
```

In [4]:
```
OHCO = ['chap_num', 'para_num', 'sent_num', 'token_num']
```

Import file into a dataframe

In [5]:
```
LINES = pd.DataFrame(open(text_file, 'r', encoding='utf-8-sig').readlines(), columns=[
LINES.index.name = 'line_num'
LINES.line_str = LINES.line_str.str.replace(r'\n+', ' ', regex=True).str.strip()
```

In [6]:
```
LINES.sample(20)
```

Out[6]:

line_num	line_str
9995	her future home by her brother and Mrs. Jennin...
3144	subject, she said to him, "Do not you know my ...
12170	choice; and she has actually been bribing one ...
4213	
10954	made it worse."
11284	her mother's presence in aid, it proceeded so ...
10541	
4130	
10624	"It is hardly worth while, Mr. Willoughby, for...
3503	wait till the door was opened before she told ...
8013	regretted being from home, when he called befo...
3110	of having often wished you to treat our acquai...
50	man, who lived to a very advanced age, and who...
803	side in the agreement; and she waited only for...
3235	She was sitting by Edward, and in taking his t...
3190	rugged; and distant objects out of sight, whic...
595	equal to your sense of his merits. I have not...
4901	
10878	rousing himself, broke it thus:
9817	

Extract Title

In [7]:
```
title = LINES.loc[0].line_str.replace('The Project Gutenberg EBook of ', '')
```

In [8]:
```
print(title)
```

```
Sense and Sensibility, by Jane Austen
```

Remove Gutenberg's front and back matter using the lines that indicate the start and end of the project.

In [9]:
```
clip_pats = [
    r"\*\*\*\s*START OF (?:THE|THIS) PROJECT",
    r"\*\*\*\s*END OF (?:THE|THIS) PROJECT"
]
```

In [10]:
```
pat_a = LINES.line_str.str.match(clip_pats[0])
pat_b = LINES.line_str.str.match(clip_pats[1])
```

In [11]:
```
line_a = LINES.loc[pat_a].index[0] + 1
line_b = LINES.loc[pat_b].index[0] - 1
```

In [12]:
```
line_a, line_b
```

Out[12]:
```
(19, 12667)
```

In [13]:
```
LINES = LINES.loc[line_a : line_b]
```

In [14]:
```
LINES.head(10)
```

Out[14]:

line_num	line_str
19	
20	Special thanks are due to Sharon Partridge for...
21	proofreading and correction of this etext.
22	
23	
24	
25	
26	
27	
28	

In [16]:
```
clip_pats = [
    r"Special"]
```

In [17]:
```
pat_a = LINES.line_str.str.match(clip_pats[0])
```

In [18]:
```
line_a = LINES.loc[pat_a].index[0] + 1
```

In [19]:
```
LINES = LINES.loc[line_a : line_b]
```

In [20]:
```
LINES.head(10)
```

Out[20]:

line_num	line_str
21	proofreading and correction of this etext.
22	
23	
24	
25	
26	
27	
28	
29	
30	

In [21]:
```
clip_pats = [
    r"proofreading"]
```

In [22]:
```
pat_a = LINES.line_str.str.match(clip_pats[0])
```

In [23]:
```
line_a = LINES.loc[pat_a].index[0] + 1
```

In [24]:
```
LINES = LINES.loc[line_a : line_b]
```

In [25]:
```
LINES.head(10)
```

Out[25]:

line_num	line_str
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	

In [15]:
```
LINES.tail(10)
```

Out[15]:

line_num	line_str
12658	
12659	
12660	
12661	
12662	
12663	
12664	
12665	
12666	End of the Project Gutenberg EBook of Sense an...
12667	

Chunk by chapter, using the pattern of locating the headers in the data frame, assigning them numbers, forward-filling those numbers, and then grouping by number (and cleaning up).

Find all chapter headers

In [26]:
```
chap_pat = r"^\s*(?:chapter|letter)\s+\d+"
```

In [27]:
```
chap_lines = LINES.line_str.str.match(chap_pat, case=False) # Returns a truth vector
```

In [28]:
```
LINES.loc[chap_lines] # Use as filter for dataframe
```

Out[28]:

line_num	line_str
42	CHAPTER 1
196	CHAPTER 2
399	CHAPTER 3
562	CHAPTER 4
757	CHAPTER 5
859	CHAPTER 6
987	CHAPTER 7
1113	CHAPTER 8
1245	CHAPTER 9
1449	CHAPTER 10
1666	CHAPTER 11
1817	CHAPTER 12
1998	CHAPTER 13
2282	CHAPTER 14
2441	CHAPTER 15
2719	CHAPTER 16
2946	CHAPTER 17
3154	CHAPTER 18
3332	CHAPTER 19
3633	CHAPTER 20
3914	CHAPTER 21
4215	CHAPTER 22
4533	CHAPTER 23
4768	CHAPTER 24
5002	CHAPTER 25
5198	CHAPTER 26
5455	CHAPTER 27
5733	CHAPTER 28
5884	CHAPTER 29
6325	CHAPTER 30
6629	CHAPTER 31
7005	CHAPTER 32
7279	CHAPTER 33

line_num	line_str
7602	CHAPTER 34
7889	CHAPTER 35
8153	CHAPTER 36
8457	CHAPTER 37
8901	CHAPTER 38
9206	CHAPTER 39
9409	CHAPTER 40
9707	CHAPTER 41
9978	CHAPTER 42
10156	CHAPTER 43
10491	CHAPTER 44
11061	CHAPTER 45
11279	CHAPTER 46
11572	CHAPTER 47
11839	CHAPTER 48
11987	CHAPTER 49
12411	CHAPTER 50

Assign numbers to chapters

In [29]:
```
LINES.loc[chap_lines, 'chap_num'] = [i+1 for i in range(LINES.loc[chap_lines].shape[0]
```

In [30]:
```
LINES.loc[chap_lines]
```

Out[30]:

line_num	line_str	chap_num
42	CHAPTER 1	1.0
196	CHAPTER 2	2.0
399	CHAPTER 3	3.0
562	CHAPTER 4	4.0
757	CHAPTER 5	5.0
859	CHAPTER 6	6.0
987	CHAPTER 7	7.0
1113	CHAPTER 8	8.0
1245	CHAPTER 9	9.0
1449	CHAPTER 10	10.0
1666	CHAPTER 11	11.0
1817	CHAPTER 12	12.0
1998	CHAPTER 13	13.0
2282	CHAPTER 14	14.0
2441	CHAPTER 15	15.0
2719	CHAPTER 16	16.0
2946	CHAPTER 17	17.0
3154	CHAPTER 18	18.0
3332	CHAPTER 19	19.0
3633	CHAPTER 20	20.0
3914	CHAPTER 21	21.0
4215	CHAPTER 22	22.0
4533	CHAPTER 23	23.0
4768	CHAPTER 24	24.0
5002	CHAPTER 25	25.0
5198	CHAPTER 26	26.0
5455	CHAPTER 27	27.0
5733	CHAPTER 28	28.0
5884	CHAPTER 29	29.0
6325	CHAPTER 30	30.0
6629	CHAPTER 31	31.0
7005	CHAPTER 32	32.0
7279	CHAPTER 33	33.0

line_num	line_str	chap_num
7602	CHAPTER 34	34.0
7889	CHAPTER 35	35.0
8153	CHAPTER 36	36.0
8457	CHAPTER 37	37.0
8901	CHAPTER 38	38.0
9206	CHAPTER 39	39.0
9409	CHAPTER 40	40.0
9707	CHAPTER 41	41.0
9978	CHAPTER 42	42.0
10156	CHAPTER 43	43.0
10491	CHAPTER 44	44.0
11061	CHAPTER 45	45.0
11279	CHAPTER 46	46.0
11572	CHAPTER 47	47.0
11839	CHAPTER 48	48.0
11987	CHAPTER 49	49.0
12411	CHAPTER 50	50.0

Notice that all lines that are not chapter headers have no chapter number assigned to them.

In [31]:
```
LINES.sample(10)
```

Out[31]:

line_num	line_str	chap_num
3963		NaN
7480	that if I had not happened to have the necessa...	NaN
4349	increased with her increase of emotion.	NaN
6477	indeed! after taking her all over Allenham Hou...	NaN
12057	satisfaction of a sleepless night. Mrs. Dashw...	NaN
5562	"You are expecting a letter, then?" said Elino...	NaN
8874	over the business."	NaN
3715	"My love you contradict every body," said his ...	NaN
2890	does at this time of the year. The woods and ...	NaN
1206		NaN

Forward-fill chapter numbers to following text lines

`ffill()` will replace null values with the previous non-null value.

In [32]:
```
LINES.chap_num = LINES.chap_num.ffill()
```

In [34]:
```
LINES.sample(10)
```

Out[34]:

line_num	line_str	chap_num
4039	similar distress last week, some apricot marma...	21.0
6196	of you; but if I am to do it, if I am to learn...	29.0
7858	affectionate sensibility, she moved after a mo...	34.0
4106		21.0
7117	them to Elinor. She could soon tell at what c...	32.0
837	money away.	5.0
10978	morning received from Mrs. Jennings declared h...	44.0
10529	hear her.	44.0
1573	which Mrs. Jennings had assigned him for her o...	10.0
8316	superiority by nature, merely from the advanta...	36.0

Notice that the lines taht precede our first chapter have no chapters, which is what we want. We need to decide whether to keep these lines as textual front matter or to dispose of them.

In [35]:
```
LINES.head(20)
```

Out[35]:

line_num	line_str	chap_num
22		NaN
23		NaN
24		NaN
25		NaN
26		NaN
27		NaN
28		NaN
29		NaN
30		NaN
31		NaN
32		NaN
33	SENSE AND SENSIBILITY	NaN
34		NaN
35	by Jane Austen	NaN
36		NaN
37	(1811)	NaN
38		NaN
39		NaN
40		NaN
41		NaN

Clean up

In [36]:
```
LINES = LINES.dropna(subset=['chap_num']) # Remove everything before Chapter 1
# LINES = LINES.loc[~LINES.chap_num.isna()] # Remove everything before Chapter 1 (alte
LINES = LINES.loc[~chap_lines] # Remove chapter heading lines; their work is done
LINES.chap_num = LINES.chap_num.astype('int') # Convert chap_num from float to int
```

In [37]:
```
LINES.sample(10)
```

Out[37]:

line_num	line_str	chap_num
7922	as Mrs. Ferrars's way of treating me yesterday...	35
5552	visit there. A note was just then brought in,...	27
4519	conversation could be continued no farther. A...	22
959	striking, and her address graceful. Her manne...	6
9419	"Thank you, ma'am," said Elinor. "It is a mat...	40
2764		16
11953		48
5085	herself, how much the heart of Marianne was in...	25
7617	The same manners, however, which recommended M...	34
2233		13

Group lines into chapters

In [38]:
```
OHCO[:1]
```

Out[38]:
```
['chap_num']
```

In [39]:
```
# Make big string for each chapter
CHAPS = LINES.groupby(OHCO[:1])\
    .line_str.apply(lambda x: '\n'.join(x))\
    .to_frame('chap_str')
```

In [40]:
```
CHAPS.head(10)
```

Out[40]:

chap_num	chap_str
1	\n\nThe family of Dashwood had long been settl...
2	\n\nMrs. John Dashwood now installed herself m...
3	\n\nMrs. Dashwood remained at Norland several ...
4	\n\n"What a pity it is, Elinor," said Marianne...
5	\n\nNo sooner was her answer dispatched, than ...
6	\n\nThe first part of their journey was perfor...
7	\n\nBarton Park was about half a mile from the...
8	\n\nMrs. Jennings was a widow with an ample jo...
9	\n\nThe Dashwoods were now settled at Barton w...
10	\n\nMarianne's preserver, as Margaret, with mo...

In [41]:
```
CHAPS['chap_str'] = CHAPS.chap_str.str.strip()
```

In [42]:
```
CHAPS
```

Out[42]:

chap_num	chap_str
1	The family of Dashwood had long been settled i...
2	Mrs. John Dashwood now installed herself mistr...
3	Mrs. Dashwood remained at Norland several mont...
4	"What a pity it is, Elinor," said Marianne, "t...
5	No sooner was her answer dispatched, than Mrs....
6	The first part of their journey was performed ...
7	Barton Park was about half a mile from the cot...
8	Mrs. Jennings was a widow with an ample jointu...
9	The Dashwoods were now settled at Barton with ...
10	Marianne's preserver, as Margaret, with more e...
11	Little had Mrs. Dashwood or her daughters imag...
12	As Elinor and Marianne were walking together t...
13	Their intended excursion to Whitwell turned ou...
14	The sudden termination of Colonel Brandon's vi...
15	Mrs. Dashwood's visit to Lady Middleton took p...
16	Marianne would have thought herself very inexc...
17	Mrs. Dashwood was surprised only for a moment ...
18	Elinor saw, with great uneasiness the low spir...
19	Edward remained a week at the cottage; he was ...
20	As the Miss Dashwoods entered the drawing-room...
21	The Palmers returned to Cleveland the next day...
22	Marianne, who had never much toleration for an...
23	However small Elinor's general dependence on L...
24	In a firm, though cautious tone, Elinor thus b...
25	Though Mrs. Jennings was in the habit of spend...
26	Elinor could not find herself in the carriage ...
27	"If this open weather holds much longer," said...
28	Nothing occurred during the next three or four...
29	Before the house-maid had lit their fire the n...
30	Mrs. Jennings came immediately to their room o...
31	From a night of more sleep than she had expect...
32	When the particulars of this conversation were...
33	After some opposition, Marianne yielded to her...

	chap_str
chap_num	
34	Mrs. John Dashwood had so much confidence in h...
35	Elinor's curiosity to see Mrs. Ferrars was sat...
36	Within a few days after this meeting, the news...
37	Mrs. Palmer was so well at the end of a fortni...
38	Mrs. Jennings was very warm in her praise of E...
39	The Miss Dashwoods had now been rather more th...
40	"Well, Miss Dashwood," said Mrs. Jennings, sag...
41	Edward, having carried his thanks to Colonel B...
42	One other short call in Harley Street, in whic...
43	Marianne got up the next morning at her usual ...
44	Elinor, starting back with a look of horror at...
45	Elinor, for some time after he left her, for s...
46	Marianne's illness, though weakening in its ki...
47	Mrs. Dashwood did not hear unmoved the vindica...
48	Elinor now found the difference between the ex...
49	Unaccountable, however, as the circumstances o...
50	After a proper resistance on the part of Mrs. ...

Split resulting data frame into paragraphs using the regex provided.

In [43]:
```
para_pat = r'\n\n+'
```

In [34]:
```
# CHAPS['chap_str'].str.split(para_pat, expand=True).head()
```

In [44]:
```
PARAS = CHAPS['chap_str'].str.split(para_pat, expand=True).stack()\
    .to_frame('para_str').sort_index()
PARAS.index.names = OHCO[:2]
```

In [45]:
```
PARAS.head()
```

Out[45]:

chap_num	para_num	para_str
1	0	The family of Dashwood had long been settled i...
	1	By a former marriage, Mr. Henry Dashwood had o...
	2	The old gentleman died: his will was read, and...
	3	Mr. Dashwood's disappointment was, at first, s...
	4	His son was sent for as soon as his danger was...

In [46]:
```
PARAS['para_str'] = PARAS['para_str'].str.replace(r'\n', ' ', regex=True)
PARAS['para_str'] = PARAS['para_str'].str.strip()
PARAS = PARAS[~PARAS['para_str'].str.match(r'^\s*$')] # Remove empty paragraphs
```

In [48]:
```
PARAS.head()
```

Out[48]:

chap_num	para_num	para_str
1	0	The family of Dashwood had long been settled i...
	1	By a former marriage, Mr. Henry Dashwood had o...
	2	The old gentleman died: his will was read, and...
	3	Mr. Dashwood's disappointment was, at first, s...
	4	His son was sent for as soon as his danger was...

Split resulting data frame into sentences using the regex provided.

In [49]:
```
# sent_pat = r'[.?!;:"]+'
sent_pat = r'[.?!;:]+'
SENTS = PARAS['para_str'].str.split(sent_pat, expand=True).stack()\
    .to_frame('sent_str')
SENTS.index.names = OHCO[:3]
```

In [50]:
```
SENTS = SENTS[~SENTS['sent_str'].str.match(r'^\s*$')] # Remove empty paragraphs
SENTS.sent_str = SENTS.sent_str.str.strip() # CRUCIAL TO REMOVE BLANK TOKENS
```

In [51]:
```
SENTS.head()
```

Out[51]:

			sent_str
chap_num	**para_num**	**sent_num**	
1	**0**	**0**	The family of Dashwood had long been settled i...
		1	Their estate was large, and their residence wa...
		2	The late owner of this estate was a single man...
		3	But her death, which happened ten years before...
		4	for to supply her loss, he invited and receive...

In [80]:
```
SENTS
```

Out[80]:

			sent_str
chap_num	**para_num**	**sent_num**	
1	**0**	**0**	The family of Dashwood had long been settled i...
		1	Their estate was large, and their residence wa...
		2	The late owner of this estate was a single man...
		3	But her death, which happened ten years before...
		4	for to supply her loss, he invited and receive...
...	**...**	**...**	...
50	**19**	**3**	Jennings, when Marianne was taken from them, M...
	20	**0**	Between Barton and Delaford, there was that co...
		1	--and among the merits and the happiness of El...
	21	**0**	THE END
	22	**0**	End of the Project Gutenberg EBook of Sense an...

8597 rows × 1 columns

In [52]:
```
SENTS.sample(10)
```

Out[52]:

chap_num	para_num	sent_num	sent_str
26	30	7	She sometimes endeavoured for a few minutes to...
23	0	5	and Edward's visit near Plymouth, his melancho...
21	40	6	Jennings deficient either in curiosity after p...
37	8	3	Ferrars would say and do, though there could n...
17	27	1	"
29	69	4	Oh, what would HE say to that
15	18	4	and the next that some unfortunate quarrel had...
49	42	3	--I can make no submission--I am grown neither...
43	19	3	and though trying to speak comfort to Elinor, ...
15	2	0	"Is anything the matter with her

Split resulting data frame into tokens using the regex provided.

In [53]:
```
token_pat = r"[\s',-]+"
TOKENS = SENTS['sent_str'].str.split(token_pat, expand=True).stack()\
    .to_frame('token_str')
```

In [54]:
```
TOKENS.index.names = OHCO[:4]
```

In [55]:
```
TOKENS
```

Out[55]:

				token_str
chap_num	para_num	sent_num	token_num	
1	0	0	0	The
			1	family
			2	of
			3	Dashwood
			4	had
...	...	...	...	...
50	22	0	8	and
			9	Sensibility
			10	by
			11	Jane
			12	Austen

122884 rows × 1 columns

Extract Vocabulary

In [56]:
```
TOKENS['term_str'] = TOKENS.token_str.replace(r'[\W_]+', '', regex=True).str.lower()
VOCAB = TOKENS.term_str.value_counts().to_frame('n').reset_index().rename(columns={'ir
VOCAB.index.name = 'term_id'
```

In [57]:
```
VOCAB
```

Out[57]:

term_id	term_str	n
0	to	4116
1	the	4106
2	of	3573
3	and	3490
4	her	2543
...	...	...
6273	prefer	1
6274	dissolving	1
6275	beset	1
6276	effectually	1
6277	austen	1

6278 rows × 2 columns

Create dataframe

In [59]:
```
old_csv_file = "C:\\Users\\ddj6tu\\Documents\\GitHub\\DS5001\\data\\austen-persuasion.
```

In [62]:
```
per = pd.read_csv(old_csv_file)
per['book_id'] = 1
```

In [63]:
```
per
```

Out[63]:

	chap_num	para_num	sent_num	token_num	token_str	term_str	book_id
0	1	0	0	0	Sir	sir	1
1	1	0	0	1	Walter	walter	1
2	1	0	0	2	Elliot	elliot	1
3	1	0	0	3	of	of	1
4	1	0	0	4	Kellynch	kellynch	1
...	...	...	...	...	...	...	...
85009	24	13	0	6	of	of	1
85010	24	13	0	7	Persuasion	persuasion	1
85011	24	13	0	8	by	by	1
85012	24	13	0	9	Jane	jane	1
85013	24	13	0	10	Austen	austen	1

85014 rows × 7 columns

In [65]:
```
TOKENS['book_id'] = 2
```

In [66]:
```
TOKENS
```

Out[66]:

				token_str	term_str	book_id
chap_num	**para_num**	**sent_num**	**token_num**			
1	**0**	**0**	**0**	The	the	2
			1	family	family	2
			2	of	of	2
			3	Dashwood	dashwood	2
			4	had	had	2
...	**...**	**...**	**...**	...	...	...
50	**22**	**0**	**8**	and	and	2
			9	Sensibility	sensibility	2
			10	by	by	2
			11	Jane	jane	2
			12	Austen	austen	2

122884 rows × 3 columns

In [71]:
```
ALL = pd.concat([per, TOKENS], ignore_index=True)
```

In [72]:
```
ALL
```

Out[72]:

	chap_num	para_num	sent_num	token_num	token_str	term_str	book_id
0	1.0	0.0	0.0	0.0	Sir	sir	1
1	1.0	0.0	0.0	1.0	Walter	walter	1
2	1.0	0.0	0.0	2.0	Elliot	elliot	1
3	1.0	0.0	0.0	3.0	of	of	1
4	1.0	0.0	0.0	4.0	Kellynch	kellynch	1
...	...	...	...	...	...	...	...
207893	NaN	NaN	NaN	NaN	and	and	2
207894	NaN	NaN	NaN	NaN	Sensibility	sensibility	2
207895	NaN	NaN	NaN	NaN	by	by	2
207896	NaN	NaN	NaN	NaN	Jane	jane	2
207897	NaN	NaN	NaN	NaN	Austen	austen	2

207898 rows × 7 columns

1. How many raw tokens are in the combined data frame?

207898

2. How many distinct terms are there in the combined data frame (i.e. how big is the vocabulary)?

8237

In [73]:
```
ALL['term_str'] = ALL.token_str.replace(r'[\W_]+', '', regex=True).str.lower()
VOCAB = ALL.term_str.value_counts().to_frame('n').reset_index().rename(columns={'index
VOCAB.index.name = 'term_id'
```

In [74]:
```
VOCAB
```

Out[74]:

term_id	term_str	n
0	the	7436
1	to	6924
2	and	6290
3	of	6145
4	her	3747
...	...	...
8232	fought	1
8233	brave	1
8234	rat	1
8235	mirrors	1
8236	surviving	1

8237 rows × 2 columns

3. How many more terms does the vocabulary of Sense and Sensibility have than that of Persuasion?

In [76]:
```
6277 - 5760
```

Out[76]:
```
517
```

In [75]:
```
per['term_str'] = per.token_str.replace(r'[\W_]+', '', regex=True).str.lower()
VOCAB = per.term_str.value_counts().to_frame('n').reset_index().rename(columns={'index
VOCAB.index.name = 'term_id'
VOCAB
```

Out[75]:

term_id	term_str	n
0	the	3330
1	to	2808
2	and	2800
3	of	2572
4	a	1595
...	...	...
5755	reins	1
5756	judiciously	1
5757	rut	1
5758	dung	1
5759	austen	1

5760 rows × 2 columns

4. What is the average number of tokens, rounded to an integer, per chapter in the corpus?

In [77]:
```
207898/(22+50)
```

Out[77]:
```
2887.472222222222
```

In [78]:
```
2887
```

Out[78]:
```
2887
```

5. What is the average number of tokens, rounded to an integer, per paragraph in the corpus?

In [81]:
```
207898/(8597+5612)
```

Out[81]:
```
14.631430783306355
```

In [82]:
```
15
```

Out[82]:
```
15
```