Build your own concordance

It took 500 Dominican munks to write the first concordance of the Latin bible, and it took Rabbi Mordecai Nathan 10 years to write the first concordance of the Hebrew bible. With Python, it only takes a matter of seconds to find words in a text, along with the surrounding words.

Run each cell in this notebook one at a time, in order. If something in one cell doesn't work right, it might be because you have overwritten a variable, so try going back and running all the previous cells again.

First run the code and check that everything works. Then, try modifying the code. Start with the first challenges, and then continue in order. Feel free to work together, and see how far you can get. The important thing is to learn, not to solve all the challenges!

In [21]:
```
# install the natural language toolkit package (nltk), which has a copy of sever
#including the King James Bible

%pip install nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in /home/ucloud/.local/lib/python3.12/site-pa
ckages (3.9.1)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from nlt
k) (8.1.6)
Requirement already satisfied: joblib in /home/ucloud/.local/lib/python3.12/site-
packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /home/ucloud/.local/lib/python
3.12/site-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /home/ucloud/.local/lib/python3.12/site-pa
ckages (from nltk) (4.67.1)
Note: you may need to restart the kernel to use updated packages.
```

In [22]:
```
# import the nltk package so that it is accessible to Python, and download a col
import nltk
nltk.download('gutenberg')
```

```
[nltk_data] Downloading package gutenberg to /home/ucloud/nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
```

Out[22]: True

In [23]:
```
# Create a variable called "bible" which contains the text of the King James bib
bible = nltk.corpus.gutenberg.raw('bible-kjv.txt')

# make all characters lowercase
bible = bible.lower()

# remove the "\n" characters, which indicate line breaks in the text (newlines)
bible = bible.replace('\n', ' ')

# split up the text into a long list of individual words
bible = bible.split(' ')
```

In [24]:
```
# make a variable called "concordance", and fill it with every occurrence of the
concordance = []
for i, val in enumerate(bible):
    if val == "world":
        if bible[i-1] == "this":
            concordance.append(str(' '.join(bible[i-5:i+5])))
```

In [25]:
```
# take a look at what the algorithm has found
concordance
```

Out[25]:
```
['for the children of this world are in their generation',
 'them, the children of this world marry, and are given',
 'hateth his life in this world shall keep it unto',
 'shall the prince of this world be cast out. ',
 'should depart out of this world unto the father, having',
 'for the prince of this world cometh, and hath nothing',
 'because the prince of this world is judged.  16:12',
 'of the princes of this world knew: for had they',
 'for the wisdom of this world is foolishness with god.',
 'for the fashion of this world passeth away.  7:32',
 'whom the god of this world hath blinded the minds',
 'chosen the poor of this world rich in faith, and',
 'saying, the kingdoms of this world are become the kingdoms']
```

In [26]:
```
# let's see how many instances of the phrase "this world" were found
len(concordance)
```

Out[26]: 13

Let's try again, but this time let's just search for "world" by itself, not "this world".

In [27]:
```
concordance = []
for i, val in enumerate(bible):
    if val == "world":
        concordance.append(str(' '.join(bible[i-5:i+5])))
```

In [28]:
```
# take a look at what the algorithm has found
concordance
```

Out[28]:
```
['and he hath set the world upon them.  2:9',
 'appeared, the foundations of the world were discovered, at the',
 'him, all the earth: the world also shall be stable,',
 'upon the face of the world in the earth. ',
 'and he shall judge the world in righteousness, he shall',
 'and the foundations of the world were discovered at thy',
 'all the ends of the world shall remember and turn',
 'all the inhabitants of the world stand in awe of',
 'not tell thee: for the world is mine, and the',
 'is thine: as for the world and the fulness thereof,',
 'he hath girded himself: the world also is stablished, that',
 'that the lord reigneth: the world also shall be established',
 'earth: he shall judge the world with righteousness, and the',
 'also he hath set the world in their heart, so',
 'and i will punish the world for their evil, and',
 'kingdoms; 14:17 that made the world as a wilderness, and',
 'fill the face of the world with cities.  14:22',
 'all the kingdoms of the world upon the face of',
 'mourneth and fadeth away, the world languisheth and fadeth away,',
 'earth, the inhabitants of the world will learn righteousness. ',
 'have the inhabitants of the world fallen.  26:19 thy',
 'fill the face of the world with fruit.  27:7',
 'not be ashamed nor confounded world without end.  45:18',
 'since the beginning of the world men have not heard,',
 'power, he hath established the world by his wisdom, and',
 'power, he hath established the world by his wisdom, and',
 'this world, neither in the world to come.  12:33',
 ' 18:7 woe unto the world because of offences! for',
 'be preached in all the world for a witness unto',
 'since the beginning of the world to this time, no,',
 'with persecutions; and in the world to come eternal life.',
 'which have been since the world began: 1:71 that we',
 'caesar augustus that all the world should be taxed. ',
 'all the kingdoms of the world in a moment of',
 'do the nations of the world seek after: and your',
 'for the children of this world are in their generation',
 'present time, and in the world to come life everlasting.',
 'them, the children of this world marry, and are given',
 'in the world, and the world was made by him,',
 'made by him, and the world knew him not. ',
 'not his son into the world to condemn the world;',
 'the world; but that the world through him might be',
 'alway ready.  7:7 the world cannot hate you; but',
 'and i speak to the world those things which i',
 'heareth.  9:32 since the world began was it not',
 'ye prevail nothing? behold, the world is gone after him.',
 'hateth his life in this world shall keep it unto',
 'shall the prince of this world be cast out. ',
 'should depart out of this world unto the father, having',
 'spirit of truth; whom the world cannot receive, because it',
 'a little while, and the world seeth me no more;',
 'unto you: not as the world giveth, give i unto',
 'for the prince of this world cometh, and hath nothing',
 ' 14:31 but that the world may know that i',
 'another.  15:18 if the world hate you, ye know',
 'were of the world, the world would love his own:',
 'of the world, therefore the world hateth you.  15:20',
 'come, he will reprove the world of sin, and of',
 'because the prince of this world is judged.  16:12',
 'weep and lament, but the world shall rejoice: and ye',
 'have peace.  in the world ye shall have tribulation:',
 'had with thee before the world was.  17:6 i',
 'them thy word; and the world hath hated them, because',
 'one in us: that the world may believe that thou',
 'in one; and that the world may know that thou',
 '17:25 o righteous father, the world hath not known thee:',
 'i suppose that even the world itself could not contain',
 'his holy prophets since the world began.  3:22 for',
 'these that have turned the world upside down are come',
 '17:24 god that made the world and all things therein,',
 'which he will judge the world in righteousness by that',
 'whom all asia and the world worshippeth.  19:28 and',
 'from the creation of the world are clearly seen, being',
 'be stopped, and all the world may become guilty before',
 'was kept secret since the world began, 16:26 but now',
 'the wisdom of god the world by wisdom knew not',
 'the foolish things of the world to confound the wise;',
 'the weak things of the world to confound the things',
 'which god ordained before the world unto our glory: 2:8',
 'of the princes of this world knew: for had they',
 'for the wisdom of this world is foolishness with god.',
 'the world? and if the world shall be judged by',
 'for the fashion of this world passeth away.  7:32',
 'eat no flesh while the world standeth, lest i make',
 'whom the ends of the world are come.  10:12',
 'whom the god of this world hath blinded the minds',
 'was in christ, reconciling the world unto himself, not imputing',
 'but the sorrow of the world worketh death.  7:11',
 'jesus christ, by whom the world is crucified unto me,',
 'from the beginning of the world hath been hid in',
 'christ jesus throughout all ages, world without end. amen. ',
 'christ jesus came into the world to save sinners; of',
 'in christ jesus before the world began, 1:10 but is',
 'cannot lie, promised before the world began; 1:3 but hath',
 'not put in subjection the world to come, whereof we',
 'and the powers of the world to come, 6:6 if',
 'in the end of the world hath he appeared to',
 'tormented; 11:38 (of whom the world was not worthy:) they',
 'chosen the poor of this world rich in faith, and',
 'tongue is a fire, a world of iniquity: so is',
 'that the friendship of the world is enmity with god?',
 'be a friend of the world is the enemy of',
 'corruption that is in the world through lust.  1:5',
 'in the flood upon the world of the ungodly; 2:6',
 'escaped the pollutions of the world through the knowledge of',
 'the water: 3:6 whereby the world that then was, being',
 'world.  2:17 and the world passeth away, and the',
 'sons of god: therefore the world knoweth us not, because',
 'not, my brethren, if the world hate you.  3:14',
 'of the world, and the world heareth them.  4:6',
 'of god, and the whole world lieth in wickedness. ',
 'saying, the kingdoms of this world are become the kingdoms',
 'was healed: and all the world wondered after the beast.']
```

In [29]:
```
# let's see how many instances of just the word "world" were found
len(concordance)
```

Out[29]: 113

Now, in the cell below, modify the code to search for a different word.

In [30]:
```
# add your modified code here and run the cell...
concordance = []
for i, val in enumerate(bible):
    if val == "slaughter":
        concordance.append(str(' '.join(bible[i-5:i+5])))
```

In [31]:
```
len(concordance)
```

Out[31]: 20

In [32]:
```
concordance
```

Out[32]:
```
['after his return from the slaughter of chedorlaomer, and of',
 'slew them with a great slaughter at gibeon, and chased',
 'hath been also a great slaughter among the people, and',
 'been now a much greater slaughter among the philistines? ',
 'as david returned from the slaughter of the philistine, abner',
 'david was returned from the slaughter of the philistine, that',
 'david was returned from the slaughter of the amalekites, and',
 'will say, there is a slaughter among the people that',
 'there was there a great slaughter that day of twenty',
 'amaziah was come from the slaughter of the edomites, that',
 'for him according to the slaughter of midian at the',
 'be renowned.  14:21 prepare slaughter for his children for',
 'he slain according to the slaughter of them that are',
 'in bozrah, and a great slaughter in the land of',
 'for the days of your slaughter and of your dispersions',
 'north, and every man a slaughter weapon in his hand;',
 'sword is drawn: for the slaughter it is furbished, to',
 'the wounded cry, when the slaughter is made in the',
 'yet breathing out threatenings and slaughter against the disciples of',
 'met abraham returning from the slaughter of the kings, and']
```

The nltk package has the full text of several other classic books. You can see what they are called by running the command in the cell below:

In [33]:
```
nltk.corpus.gutenberg.fileids()
```

Out[33]:
```
['austen-emma.txt',
 'austen-persuasion.txt',
 'austen-sense.txt',
 'bible-kjv.txt',
 'blake-poems.txt',
 'bryant-stories.txt',
 'burgess-busterbrown.txt',
 'carroll-alice.txt',
 'chesterton-ball.txt',
 'chesterton-brown.txt',
 'chesterton-thursday.txt',
 'edgeworth-parents.txt',
 'melville-moby_dick.txt',
 'milton-paradise.txt',
 'shakespeare-caesar.txt',
 'shakespeare-hamlet.txt',
 'shakespeare-macbeth.txt',
 'whitman-leaves.txt']
```

Your turn!

Here are a some more things you can try. In each case, I have given you a little bit of starter code to get you going, but the cells will not run without some additional code from you.

Challenge 1: build your own concordance

Pick a different book and a different word, or pair of words. Copy and paste from the code above to write some Python code that searches the book of your choice for the word or pair of words of your choice. Put this code in the cell below. By the way, some of the texts use the characters "\r" for "carriage return" instead of "\n" for "newline". You can remove these the same way that you remove the "\n" characters.

In [34]:
```
# Create a variable called "bible" which contains the text of the King James bib
shakey = nltk.corpus.gutenberg.raw('shakespeare-caesar.txt')

# make all characters lowercase
shakey = shakey.lower()

# remove the "\n" characters, which indicate line breaks in the text (newlines)
shakey = shakey.replace('\n', ' ')

# split up the text into a long list of individual words
shakey = shakey.split(' ')
```

In [35]:
```
# add your modified code here and run the cell...
shakey_concordance = []
for i, val in enumerate(shakey):
    if val == "seek":
        shakey_concordance.append(str(' '.join(shakey[i-5:i+5])))
```

In [36]:
```
shakey_concordance
```

Out[36]: ['to maske thy monstrous visage? seek none conspiracie, hide it']

In [37]:
```
len(shakey_concordance)
```

Out[37]: 1

Challenge 2: compare lengths of books

We can use the command `len` to find how many items there are in a list. E.g., to find the number of words in the list called `bible`, above, we can write: `len(bible)`.

Use the starter code below to find out which book in the books included in `nltk` has the most words.

In [38]:
```
# solution 1: print all the titles and numbers of words
# starter code:

import nltk

# Ensure the Gutenberg corpus is downloaded
nltk.download('gutenberg')

# Get the list of book titles
books = nltk.corpus.gutenberg.fileids()

books = nltk.corpus.gutenberg.fileids()
print("Unsorted Book Lengths")
for title in books:
    book = nltk.corpus.gutenberg.raw(title)
    print(title, " | ", len(book))
```

```
Unsorted Book Lengths
austen-emma.txt  |  887071
austen-persuasion.txt  |  466292
austen-sense.txt  |  673022
bible-kjv.txt  |  4332554
blake-poems.txt  |  38153
bryant-stories.txt  |  249439
burgess-busterbrown.txt  |  84663
carroll-alice.txt  |  144395
chesterton-ball.txt  |  457450
chesterton-brown.txt  |  406629
chesterton-thursday.txt  |  320525
edgeworth-parents.txt  |  935158
melville-moby_dick.txt  |  1242990
milton-paradise.txt  |  468220
shakespeare-caesar.txt  |  112310
shakespeare-hamlet.txt  |  162881
shakespeare-macbeth.txt  |  100351
whitman-leaves.txt  |  711215
```

```
[nltk_data] Downloading package gutenberg to /home/ucloud/nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
```

In [39]:
```
# more advanced, for those with some Python experience, or those who want to goo
# solution 2: make a list of titles and a list of wordcounts, zip them together,
# starter code:

books = nltk.corpus.gutenberg.fileids()

titles = []
numwords = []

for title in books:

    book = nltk.corpus.gutenberg.raw(title)

    book_tuple = (title, len(book))

    numwords.append(book_tuple)


numwords.sort(key=lambda x: x[1])


print("Sorted book lengths (ascending order):")
for title, length in numwords:
    print(f"{title}: {length}")
```

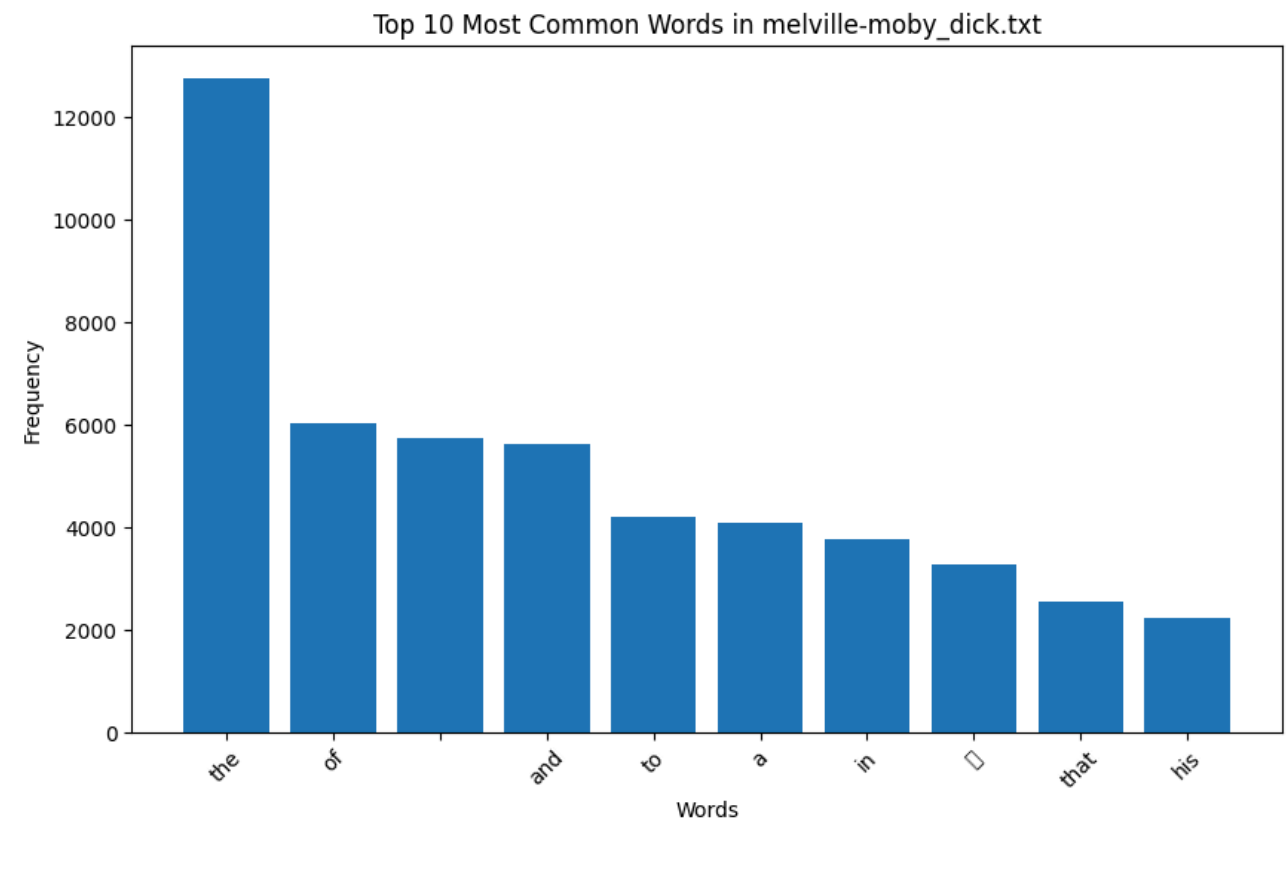
```
Sorted book lengths (ascending order):
blake-poems.txt: 38153
burgess-busterbrown.txt: 84663
shakespeare-macbeth.txt: 100351
shakespeare-caesar.txt: 112310
carroll-alice.txt: 144395
shakespeare-hamlet.txt: 162881
bryant-stories.txt: 249439
chesterton-thursday.txt: 320525
chesterton-brown.txt: 406629
chesterton-ball.txt: 457450
austen-persuasion.txt: 466292
milton-paradise.txt: 468220
austen-sense.txt: 673022
whitman-leaves.txt: 711215
austen-emma.txt: 887071
edgeworth-parents.txt: 935158
melville-moby_dick.txt: 1242990
bible-kjv.txt: 4332554
```

Challenge 3: what are the most frequent words?

`nltk` has a built-in function called `FreqDist` which counts up how many times each word in a text occurs. So, if you have a list called `words` which contains all the words in a book, you can find the frequencies of all of them by writing `freq = nltk.FreqDist(words)`. You can then get the e.g. ten most common words by writing `freq.most_common(10)`. What are the ten most common words in Jane Austen's "Emma"? What about Herman Melville's "Moby Dick"?

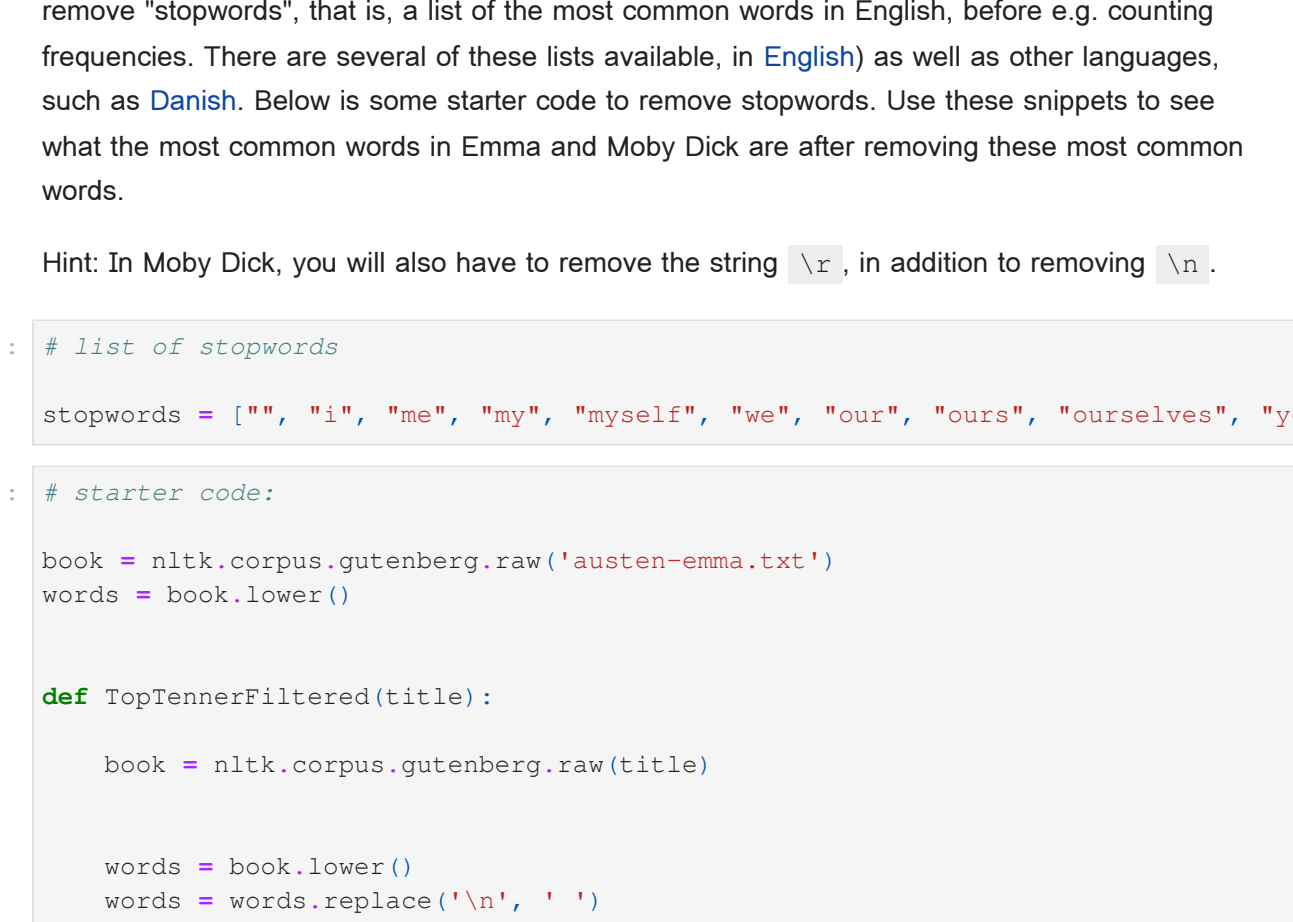
In [40]:
```
import matplotlib.pyplot as plt


nltk.download('gutenberg')

def TopTenner(title):

    book = nltk.corpus.gutenberg.raw(title)


    words = book.lower()
    words = words.replace('\n', ' ')
    words = words.split(' ')


    freq = nltk.FreqDist(words)


    most_common_words = freq.most_common(10)

    words, counts = zip(*most_common_words)

    print("For text: ", title)

    plt.figure(figsize=(10, 6))
    plt.bar(words, counts)
    plt.xlabel('Words')
    plt.ylabel('Frequency')
    plt.title(f'Top 10 Most Common Words in {title}')
    plt.xticks(rotation=45)
    plt.show()

# Example usage
TopTenner('austen-emma.txt')
```

```
[nltk_data] Downloading package gutenberg to /home/ucloud/nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
For text:  austen-emma.txt
```



In [41]:
```
TopTenner("melville-moby_dick.txt")
```

```
For text:  melville-moby_dick.txt
) missing from font(s) DejaVu Sans.site-packages/IPython/core/pylabtools.py:170:
UserWarning: Glyph 13 (
  fig.canvas.print_figure(bytes_io, **kw)
```



Challenge 4: Remove stopwords

Often, the most frequent words are not the most interesting ones. Words like "a" and "the" are so common in English, that they don't really tell us much about the text. That is why we often remove "stopwords", that is, a list of the most common words in English, before e.g. counting frequencies. There are several of these lists available, in English) as well as other languages, such as Danish. Below is some starter code to remove stopwords. Use these snippets to see what the most common words in Emma and Moby Dick are after removing these most common words.

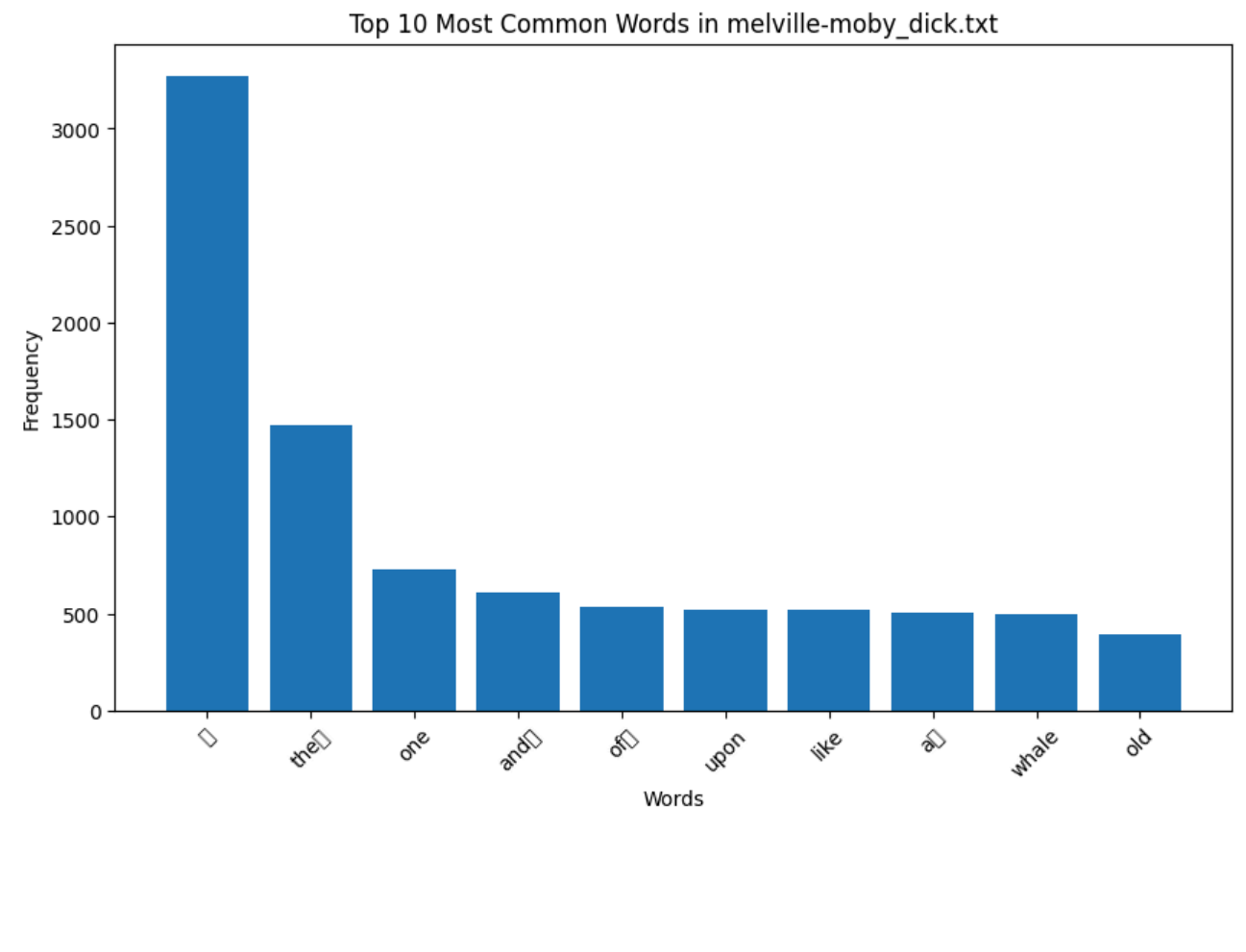
Hint: In Moby Dick, you will also have to remove the string `\r`, in addition to removing `\n`.

In [42]:
```
# list of stopwords

stopwords = ["", "i", "me", "my", "myself", "we", "our", "ours", "ourselves", "y
```

In [43]:
```
# starter code:

book = nltk.corpus.gutenberg.raw('austen-emma.txt')
words = book.lower()


def TopTennerFiltered(title):

    book = nltk.corpus.gutenberg.raw(title)


    words = book.lower()
    words = words.replace('\n', ' ')
    words = words.split(' ')

    # code to remove stopwords.
    words = [word for word in words if word not in stopwords]

    freq = nltk.FreqDist(words)


    most_common_words = freq.most_common(10)

    words, counts = zip(*most_common_words)

    print("For text: ", title)

    plt.figure(figsize=(10, 6))
    plt.bar(words, counts)
    plt.xlabel('Words')
    plt.ylabel('Frequency')
    plt.title(f'Top 10 Most Common Words in {title}')
    plt.xticks(rotation=45)
    plt.show()
```

In [44]:
```
TopTennerFiltered('austen-emma.txt')
```

```
For text:  austen-emma.txt
```



In [45]:
```
TopTennerFiltered('melville-moby_dick.txt')
```

```
For text:  melville-moby_dick.txt
```