In [6]:
```
file = open('pg37106.txt',encoding='utf-8')
```

In [7]:
```
data = file.read().splitlines()
```

In [8]:
```
data[:10]
```

Out[8]:
```
['\ufeffThe Project Gutenberg EBook of Little Women, by Louisa M. Alcott',
 '',
 'This eBook is for the use of anyone anywhere at no cost and with',
 'almost no restrictions whatsoever.  You may copy it, give it away or',
 're-use it under the terms of the Project Gutenberg License included',
 'with this eBook or online at www.gutenberg.org',
 '',
 '',
 'Title: Little Women',
 '       or Meg, Jo, Beth, and Amy']
```

Data Preprocessing

In [11]:
```
import nltk
```

In [12]:
```
from nltk.corpus import stopwords
import warnings
warnings.filterwarnings('ignore')
```

In [16]:
```
stop_words = set(stopwords.words('english'))
```

In [17]:
```
stop_words
```

Out[17]:
```
{'a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'ain',
 'all',
 'am',
 'an',
 'and',
 'any',
 'are',
 'aren',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 'couldn',
 "couldn't",
 'd',
 'did',
 'didn',
```

```
"didn't",
'do',
'does',
'doesn',
"doesn't",
'doing',
'don',
"don't",
'down',
'during',
'each',
'few',
'for',
'from',
'further',
'had',
'hadn',
"hadn't",
'has',
'hasn',
"hasn't",
'have',
'haven',
"haven't",
'having',
'he',
'her',
'here',
'hers',
'herself',
'him',
'himself',
'his',
'how',
'i',
'if',
'in',
'into',
'is',
'isn',
"isn't",
'it',
"it's",
'its',
'itself',
'just',
'll',
'm',
'ma',
'me',
'mightn',
"mightn't",
'more',
'most',
'mustn',
"mustn't",
'my',
'myself',
'needn',
"needn't",
'no',
'nor',
'not',
'now',
'o',
'of',
```

```
'off',
'on',
'once',
'only',
'or',
'other',
'our',
'ours',
'ourselves',
'out',
'over',
'own',
're',
's',
'same',
'shan',
"shan't",
'she',
"she's",
'should',
"should've",
'shouldn',
"shouldn't",
'so',
'some',
'such',
't',
'than',
'that',
"that'll",
'the',
'their',
'theirs',
'them',
'themselves',
'then',
'there',
'these',
'they',
'this',
'those',
'through',
'to',
'too',
'under',
'until',
'up',
've',
'very',
'was',
'wasn',
"wasn't",
'we',
'were',
'weren',
"weren't",
'what',
'when',
'where',
'which',
'while',
'who',
'whom',
'why',
'will',
'with',
```

```
 'won',
 "won't",
 'wouldn',
 "wouldn't",
 'y',
 'you',
 "you'd",
 "you'll",
 "you're",
 "you've",
 'your',
 'yours',
 'yourself',
 'yourselves'}
```

Converting all words into lower case to avoid discrepancies

In [14]:
```
for i in range(len(data)):
    data[i] =data[i].lower()
```

In [15]:
```
data[:10]
```

Out[15]:
```
['\ufeffthe project gutenberg ebook of little women, by louisa m. alcott',
 '',
 'this ebook is for the use of anyone anywhere at no cost and with',
 'almost no restrictions whatsoever.  you may copy it, give it away or',
 're-use it under the terms of the project gutenberg license included',
 'with this ebook or online at www.gutenberg.org',
 '',
 '',
 'title: little women',
 '       or meg, jo, beth, and amy']
```

Removing stop words

In [18]:
```
data_1= []
for i in range(len(data)):
    sentence=data[i]
    tokens = sentence.split()
     # remove stopwords
    filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

    filtered_sentence = ' '.join(filtered_tokens)
    data_1.append(filtered_sentence)
```

In [19]:
```
data_1[:10]
```

Out[19]:
```
['\ufeffthe project gutenberg ebook little women, louisa m. alcott',
 '',
 'ebook use anyone anywhere cost',
 'almost restrictions whatsoever. may copy it, give away',
 're-use terms project gutenberg license included',
 'ebook online www.gutenberg.org',
 '',
 '',
 'title: little women',
 'meg, jo, beth, amy']
```

Removing special characters

In [20]:
```
import re
```

In [21]:
```
data_2=[]
```

```
for i in range(len(data_1)):
    sentence=data_1[i]
    clean_sentence = re.sub(r'\W+', ' ', sentence)
    data_2.append(clean_sentence)
```

In [22]:
```
data_2[:10]
```

Out[22]:
```
[' the project gutenberg ebook little women louisa m alcott',
 '',
 'ebook use anyone anywhere cost',
 'almost restrictions whatsoever may copy it give away',
 're use terms project gutenberg license included',
 'ebook online www gutenberg org',
 '',
 '',
 'title little women',
 'meg jo beth amy']
```

In [24]:
```
pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.8.2.2-cp39-cp39-macosx_10_9_x86_64.whl (160 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 160.5/160.5 kB 3.0 MB/s eta 0:00:00a 0:00:0
1
Requirement already satisfied: pillow in /Users/devnaramesh/opt/anaconda3/lib/python3.9/
site-packages (from wordcloud) (9.0.1)
Requirement already satisfied: numpy>=1.6.1 in /Users/devnaramesh/opt/anaconda3/lib/pyth
on3.9/site-packages (from wordcloud) (1.21.6)
Requirement already satisfied: matplotlib in /Users/devnaramesh/opt/anaconda3/lib/python
3.9/site-packages (from wordcloud) (3.5.2)
Requirement already satisfied: fonttools>=4.22.0 in /Users/devnaramesh/opt/anaconda3/li
b/python3.9/site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: cycler>=0.10 in /Users/devnaramesh/opt/anaconda3/lib/pyth
on3.9/site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/devnaramesh/opt/anaconda3/li
b/python3.9/site-packages (from matplotlib->wordcloud) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in /Users/devnaramesh/opt/anaconda3/
lib/python3.9/site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: packaging>=20.0 in /Users/devnaramesh/opt/anaconda3/lib/p
ython3.9/site-packages (from matplotlib->wordcloud) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/devnaramesh/opt/anaconda3/lib/
python3.9/site-packages (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: six>=1.5 in /Users/devnaramesh/opt/anaconda3/lib/python3.
9/site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.2.2
Note: you may need to restart the kernel to use updated packages.
```

In [25]:
```
from wordcloud import WordCloud
```

Visualization of bag of words

In [26]:
```
import matplotlib.pyplot as plt
```

In [27]:
```
bag_of_words = " ".join(x for x in data_2)

# Generate the word cloud
wordcloud = WordCloud(width=800, height=800, background_color='white').generate(bag_of_w

# Visualize the word cloud
plt.figure(figsize=(8, 8))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

In [28]:
```
data_3= []
list_words=['jo','amy','meg','beth','girl','little','one','laurie','john','said','good',
for i in range(len(data_2)):
    sentence=data_2[i]
    tokens = sentence.split()  # split sentence into individual words
    filtered_tokens = [word for word in tokens if word.lower() not in list_words]  # rem

    filtered_sentence = ' '.join(filtered_tokens)  # join the filtered tokens back into
    data_3.append(filtered_sentence)
```

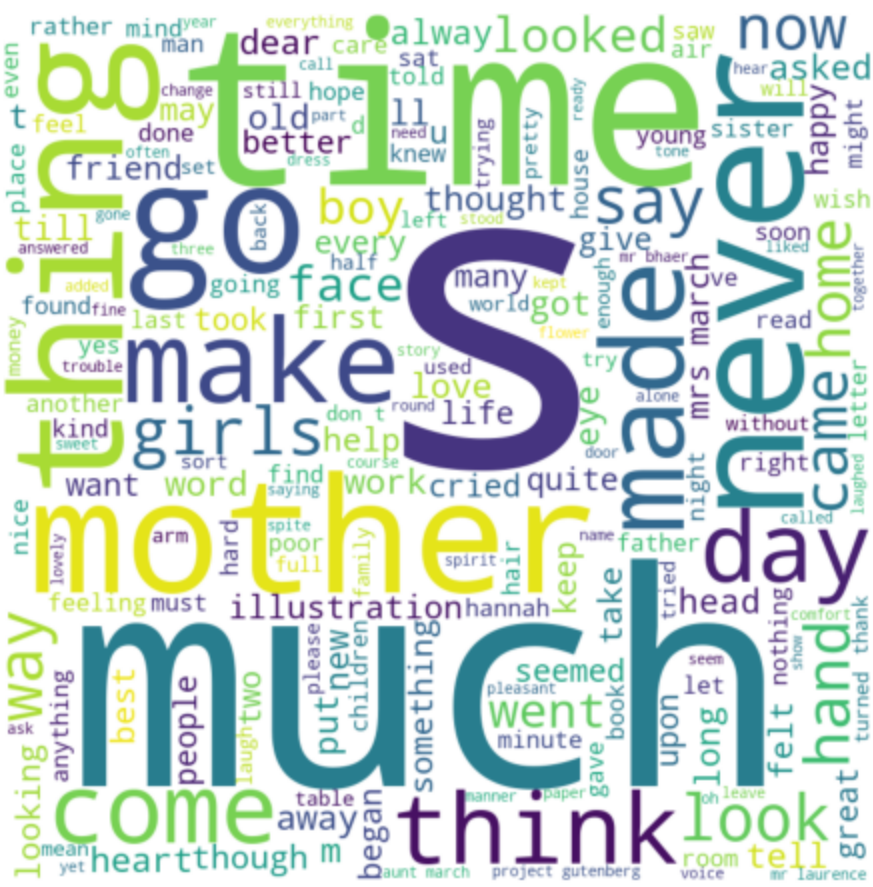
In [29]:
```
data_3[:10]
```

Out[29]:
```
['the project gutenberg ebook women louisa m alcott',
 '',
 'ebook use anyone anywhere cost',
 'almost restrictions whatsoever may copy it give away',
 're use terms project gutenberg license included',
 'ebook online www gutenberg org',
 '',
 '',
 'title women',
 '']
```

In [30]:
```
bag_of_words = " ".join(x for x in data_3)

# Generate the word cloud
wordcloud = WordCloud(width=800, height=800, background_color='white').generate(bag_of_w

# Visualize the word cloud
plt.figure(figsize=(8, 8))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



In [35]:
```
from nltk.tag import pos_tag
from nltk.tokenize import word_tokenize
from collections import Counter
```

Checking the most occuring adjectives

In [37]:
```
all_adjectives = []

# Loop through each sentence and extract all adjectives
for sentence in data_3:
    tokens = word_tokenize(sentence)
    tags = pos_tag(tokens)
    adjectives = [word for word, pos in tags if pos == 'JJ']
    all_adjectives.extend(adjectives)

# Count the frequency of each adjective
adjective_counts = Counter(all_adjectives)

# Print the top 10 most common adjectives
print(adjective_counts.most_common(10))
```

```
[('i', 502), ('old', 396), ('s', 395), ('young', 286), ('much', 241), ('new', 200), ('great', 190), ('happy', 170), ('poor', 165), ('many', 151)]
```

In [51]:
```
words=[]
for sentence in data_2:
    tokens = word_tokenize(sentence)
    words.extend(tokens)

word_freq = Counter(words)

# Get the top 10 most common words and their frequencies
top_words = word_freq.most_common(10)
top_words.reverse()

# Plot the top words as a bar graph
```

```
plt.barh(range(len(top_words)), [freq for (word, freq) in top_words], align='center')
plt.yticks(range(len(top_words)), [word for (word, freq) in top_words])
plt.xlabel("Frequency")
plt.ylabel("Word")
plt.title("Most Common Words in Little Women")
plt.show()
```