In [1]:

```
#word cloud -Also practical-8
```

In [2]:

```
import pandas as pd
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import nltk
from nltk import masi_distance
from nltk.probability import FreqDist
import urllib.request
from wordcloud import WordCloud
nltk.download('punkt')
#%matplotlib inline
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[2]:

```
True
```

In [5]:

```
#install wordcloud
#note that ! mark is required before pip
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\hp\anaconda3\lib\site
-packages (1.9.2)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\sit
e-packages (from wordcloud) (3.7.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\hp\anaconda3\lib\s
ite-packages (from wordcloud) (1.23.5)
Requirement already satisfied: pillow in c:\users\hp\anaconda3\lib\site-pa
ckages (from wordcloud) (9.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\anaconda3\li
b\site-packages (from matplotlib->wordcloud) (22.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anacond
a3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\anaconda3\l
ib\site-packages (from matplotlib->wordcloud) (1.0.5)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\s
ite-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hp\anaconda3\l
ib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
```

In [6]:

```
#open the file and read it into a variable
agatha_novel = open('C:/Users/HP/OneDrive/Desktop/DataScience/crooked-house.txt','r').re
agatha_novel[1000:2000]
```

Out[6]:

```
'ortened, and the vocabulary and grammar simplified\nto make it accessible
to readers with a good intermediate\nknowledge of the language.\nThe follo
wing features are included after the story:\nA List of characters to help
the reader identify who is who, and\nhow they are connected to each other.
Cultural notes to explain\nhistorical and other references. A Glossary of
words that some\nreaders may not be familiar with are explained. There is
also a\nRecording of the story.\n\n\x0cAgatha Christie\nCrooked House\n\nC
ollins\n\n\x0cCollins\nHarperCollins Publishers\n77-85 Fulham Palace Road
\nLondon W6 8JB\nwww.collinselt.com\n\nContents\nStory\n\n1\n\nCharacter l
ist\n\n99\n\nC ultural notes\n\n100\n\nGlossary\n\n104\n\nCollinsÂ® is a r
egistered trademark ofHarperCollins Publishers Limited.\nThis Colli11s Eng
lish Readers edition published 2012\nReprint 10 9 8 7 6 5 4 3 2 1 0\nFirst
published in Great Britain by Collins 1949\nAGATHA CHRISTIEâ„¢ Crooked Hou
seâ„¢\nCopyright Â© 1949 Agatha Christie Limited. All rights reserved.\nCo
pyright Â© 20'
```

In [4]:

```
print(agatha_novel[1000:2000])
```

```
ortened, and the vocabulary and grammar simplified
to make it accessible to readers with a good intermediate
knowledge of the language.
The following features are included after the story:
A List of characters to help the reader identify who is who, and
how they are connected to each other. Cultural notes to explain
historical and other references. A Glossary of words that some
readers may not be familiar with are explained. There is also a
Recording of the story.

Agatha Christie
Crooked House

Collins

Collins
HarperCollins Publishers
77-85 Fulham Palace Road
London W6 8JB
www.collinselt.com

Contents
Story

1

Character list

99

C ultural notes

100

Glossary

104

CollinsÂ® is a registered trademark ofHarperCollins Publishers Limited.
This Colli11s English Readers edition published 2012
Reprint 10 9 8 7 6 5 4 3 2 1 0
First published in Great Britain by Collins 1949
AGATHA CHRISTIEâ„¢ Crooked Houseâ„¢
Copyright Â© 1949 Agatha Christie Limited. All rights reserved.
Copyright Â© 20
```

In [7]:

```
from nltk import word_tokenize
words = word_tokenize(agatha_novel)
print(words)
```

```
'most', 'popular', 'detective', 'in', 'crime', 'fiction', 'since', 'She
rlock', 'Holmes', '.', 'Collins', 'has', 'published', 'Agatha', 'Christ
ie', 'since', '1926', '.', 'This', 'series', 'has', 'been', 'especiall
y', 'created', 'for', 'readers', 'worldwide', 'whose', 'first', 'langua
ge', 'is', 'not', 'English', '.', 'Each', 'story', 'has', 'been', 'shor
tened', ',', 'and', 'the', 'vocabulary', 'and', 'grammar', 'simplifie
d', 'to', 'make', 'it', 'accessible', 'to', 'readers', 'with', 'a', 'go
od', 'intermediate', 'knowledge', 'of', 'the', 'language', '.', 'The',
'following', 'features', 'are', 'included', 'after', 'the', 'story',
':', 'A', 'List', 'of', 'characters', 'to', 'help', 'the', 'reader', 'i
dentify', 'who', 'is', 'who', ',', 'and', 'how', 'they', 'are', 'connec
ted', 'to', 'each', 'other', '.', 'Cultural', 'notes', 'to', 'explain',
'historical', 'and', 'other', 'references', '.', 'A', 'Glossary', 'of',
'words', 'that', 'some', 'readers', 'may', 'not', 'be', 'familiar', 'wi
th', 'are', 'explained', '.', 'There', 'is', 'also', 'a', 'Recording',
'of', 'the', 'story', '.', 'Agatha', 'Christie', 'Crooked', 'House', 'C
ollins', 'Collins', 'HarperCollins', 'Publishers', '77-85', 'Fulham',
'Palace', 'Road', 'London', 'W6', '8JB', 'www.collinselt.com', 'Content
s', 'Story', '1', 'Character', 'list', '99', 'C', 'ultural', 'notes',
'100', 'Glossary', '104', 'CollinsÂ®', 'is', 'a', 'registered', 'tradem
```

In [8]:

```
#check number of words
len(words)
```

Out[8]:

```
38859
```

In [9]:

```
#find frequency of words
fdist = FreqDist(words)
#print the 10 most comman words
fdist.most_common(10)
```

Out[9]:

```
[('.', 2298),
 (',', 1793),
 ("'", 1643),
 ('I', 1195),
 ('the', 956),
 ('to', 750),
 ('and', 691),
 ('a', 596),
 ('was', 495),
 ('said', 449)]
```

In [10]:

```
#create an empty List to stare words
words_no_punc = []

#iterate through the words list to remove punctuations and numbers
for word in words:
    if word.isalpha():
        words_no_punc.append(word.lower())

#print nimber of words without punctuations
print(f"The Total number of words without punctions is {len(words_no_punc)}")
```

```
The Total number of words without punctions is 29534
```

In [11]:

```
#Download and import List of stopwords
nltk.download('stopwords')
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [12]:

```
#List of Stopwords
stopwords_list = stopwords.words('english')
print(stopwords_list)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "yo
u're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselv
es', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'hersel
f', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'the
mselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'thes
e', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'hav
e', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'th
e', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'a
t', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through',
'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',
'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'on
ce', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'no
t', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can',
'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll',
'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",
'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't",
'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn',
"mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't",
'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "would
n't"]
```

In [13]:

```
#create an empty list to store clean words
clean_words = []

##iterate through the words_no_punc list and add non stopwords to the new clean_words li
for word in words_no_punc:
    if word not in stopwords_list:
        clean_words.append(word)

print(f"The Total number of words without punctions and stopwords is {len(clean_words)}
```

```
The Total number of words without punctions and stopwords is 14936
```

In [14]:

```
print(clean_words)
```

```
'included', 'story', 'list', 'characters', 'help', 'reader', 'identif
y', 'connected', 'cultural', 'notes', 'explain', 'historical', 'referen
ces', 'glossary', 'words', 'readers', 'may', 'familiar', 'explained',
'also', 'recording', 'story', 'agatha', 'christie', 'crooked', 'house',
'collins', 'collins', 'harpercollins', 'publishers', 'fulham', 'palac
e', 'road', 'london', 'contents', 'story', 'character', 'list', 'c', 'u
ltural', 'notes', 'glossary', 'registered', 'trademark', 'ofharpercolli
ns', 'publishers', 'limited', 'english', 'readers', 'edition', 'publish
ed', 'reprint', 'first', 'published', 'great', 'britain', 'collins', 'a
gatha', 'christieâ', 'crooked', 'houseâ', 'copyright', 'agatha', 'chris
tie', 'limited', 'rights', 'reserved', 'copyright', 'crooked', 'house
â', 'abridged', 'edition', 'agatha', 'c', 'hristie', 'limited', 'right
s', 'reserved', 'isbn', 'catalogue', 'record', 'book', 'available', 'br
itish', 'library', 'cover', 'c', 'agatha', 'christie', 'ltd', 'typese
t', 'aptara', 'india', 'printed', 'bound', 'great', 'britain', 'clays',
'ltd', 'st', 'ives', 'pie', 'rights', 'reserved', 'part', 'publicatio
n', 'may', 'reproduced', 'stored', 'retrieval', 'system', 'transmitte
d', 'form', 'means', 'electronic', 'mechanical', 'photocopying', 'recor
ding', 'otherwise', 'without', 'prior', 'permission', 'publishers', 'bo
ok', 'sold', 'subject', 'condition', 'shall', 'way', 'rade', 'ul', 'ud
```

In [15]:

```
#find the frequency of words
fdist = FreqDist(clean_words)
fdist.most_common(10)
```

Out[15]:

```
[('said', 449),
 ('sophia', 199),
 ('father', 156),
 ('leonides', 145),
 ('josephine', 139),
 ('taverner', 136),
 ('house', 127),
 ('roger', 127),
 ('know', 125),
 ('think', 119)]
```

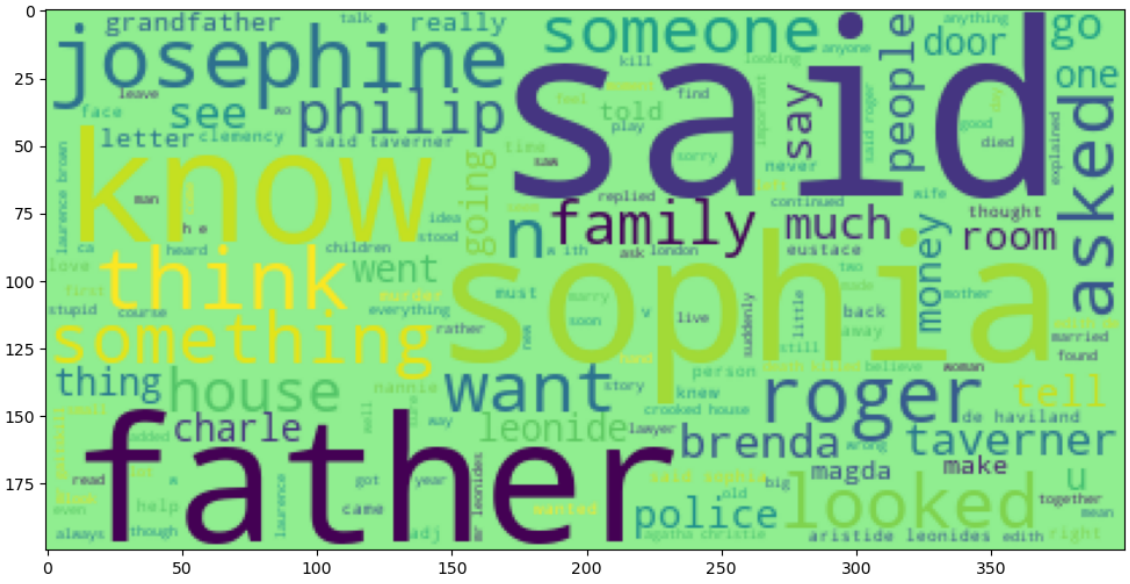
In [17]:

```
#instantiate a word cloud object
#convert word list to a single string
clean_words_string = " ".join(clean_words)

#generating the wordcloud
wordcloud = WordCloud(background_color="lightgreen").generate(clean_words_string)

#plot the wordcloud
plt.figure(figsize=(12,12))
plt.imshow(wordcloud)

#to remove the axis value
#plt.axis("off")
plt.show()
```



In [ ]:

```
```