Wikibook title and text generation using LSTM

Created by Nicholas Felcher

Github: https://github.com/NicholasFelcher/CISB63_Final (https://github.com/NicholasFelcher/CISB63_Final)

Dataset: https://www.kaggle.com/datasets/dhruvildave/wikibooks-dataset/ (https://www.kaggle.com/datasets/dhruvildave/wikibooks-dataset/)

Import required libraries

In [286]:
```
#data visualization
import pandas as pd
import numpy as np
import string
import os
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt


#warnings
import warnings
warnings.filterwarnings("ignore")

#database retrieval
import sqlite3

#nltk, stemmers, spacy, regex
import nltk
import spacy
import re

#LSTM, deep learning
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Embedding, LSTM, Dense, Dropout
from keras.preprocessing.text import Tokenizer
from keras.callbacks import EarlyStopping
from keras.models import Sequential
import keras.utils as ku
```

In [3]:
```
#connect to the database
conn = sqlite3.connect('wikibooks.sqlite')
df = pd.read_sql_query("SELECT * FROM en", conn)
```

In [4]:
```
conn.close()
```

EDA

In [5]:
```
df.head()
```

Out[5]:

	title	url	abstract	body_text	body_html
0	Wikibooks: Radiation Oncology/NHL/CLL-SLL	https://en.wikibooks.org/wiki/Radiation_Oncolo...	Chronic Lymphocytic Leukemia and Small Lymphoc...	Front Page: Radiation Oncology | RTOG Trials |...	<table width="10...
1	Wikibooks: Romanian/Lesson 9	https://en.wikibooks.org/wiki/Romanian/Lesson_9	==Băuturi/Beverages==	Băuturi/Beverages[edit | edit source]\nTea : C...	<h2><span id="B....
2	Wikibooks: Karrigell	https://en.wikibooks.org/wiki/Karrigell	Karrigell is an open Source Python web framewo...	Karrigell is an open Source Python web framewo...	<p>Karrigell is ...
3	Wikibooks: The Pyrogenesis Engine/0 A.D./GuiSe...	https://en.wikibooks.org/wiki/The_Pyrogenesis_...	====setupUnitPanel====	setupUnitPanel[edit | edit source]\nHelper fun...	<h4><span class=...
4	Wikibooks: LMIs in Control/pages/Exterior Coni...	https://en.wikibooks.org/wiki/LMIs_in_Control/...	== The Concept ==	Contents\n\n1 The Concept\n2 The System\n3 The...	<div id="toc" cl...

In [6]:
```
df.describe()
```

Out[6]:

	title	url	abstract	body_text	body_html
count	86736	86736	86736	86736	86736
unique	86736	86736	62141	85318	86736
top	Wikibooks: Radiation Oncology/NHL/CLL-SLL	https://en.wikibooks.org/wiki/Radiation_Oncolo...		Annotations[edit | edit source]\nReferences[ed...	<table width="10...
freq	1	1	2130	362	1

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86736 entries, 0 to 86735
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   title      86736 non-null  object
 1   url        86736 non-null  object
 2   abstract   86736 non-null  object
 3   body_text  86736 non-null  object
 4   body_html  86736 non-null  object
dtypes: object(5)
memory usage: 3.3+ MB
```

No null values, so cleaning is easy

Cleaning/Preprocessing Titles

In [8]:
```
df['title'][3]
```

Out[8]:
```
'Wikibooks: The Pyrogenesis Engine/0 A.D./GuiSession'
```

In [9]:
```
nlp = spacy.load('en_core_web_sm')
#cleans the titles by removing constant elements, punctuation, uppercase words, and stopwords.
def clean_title(txt, stem="None"):
    final_string = ""
    #Remove the Constant element in each title
    txt = txt.replace('Wikibooks: ' ,'')
    #remove uppercase
    txt = txt.lower()
    #remove punctuation
    for i in string.punctuation:
        txt = txt.replace(i ,' ')
    #split txt
    txt = txt.split()
    #retrieve list of stopwords
    stop_words = nltk.corpus.stopwords.words("english")
    text_filtered = [word for word in txt if not word in stop_words]

    final_string = ' '.join(text_filtered)
    return final_string
```

In [10]:
```
df['title_clean'] = df['title'].apply(lambda x: clean_title(x))
```

In [95]:
```
df['title_clean'].head()
```

Out[95]:
```
0                  radiation oncology nhl cll sll
1                              romanian lesson 9
2                                      karrigell
3                pyrogenesis engine 0 guisession
4    lmis control pages exterior conic sector lemma
Name: title_clean, dtype: object
```

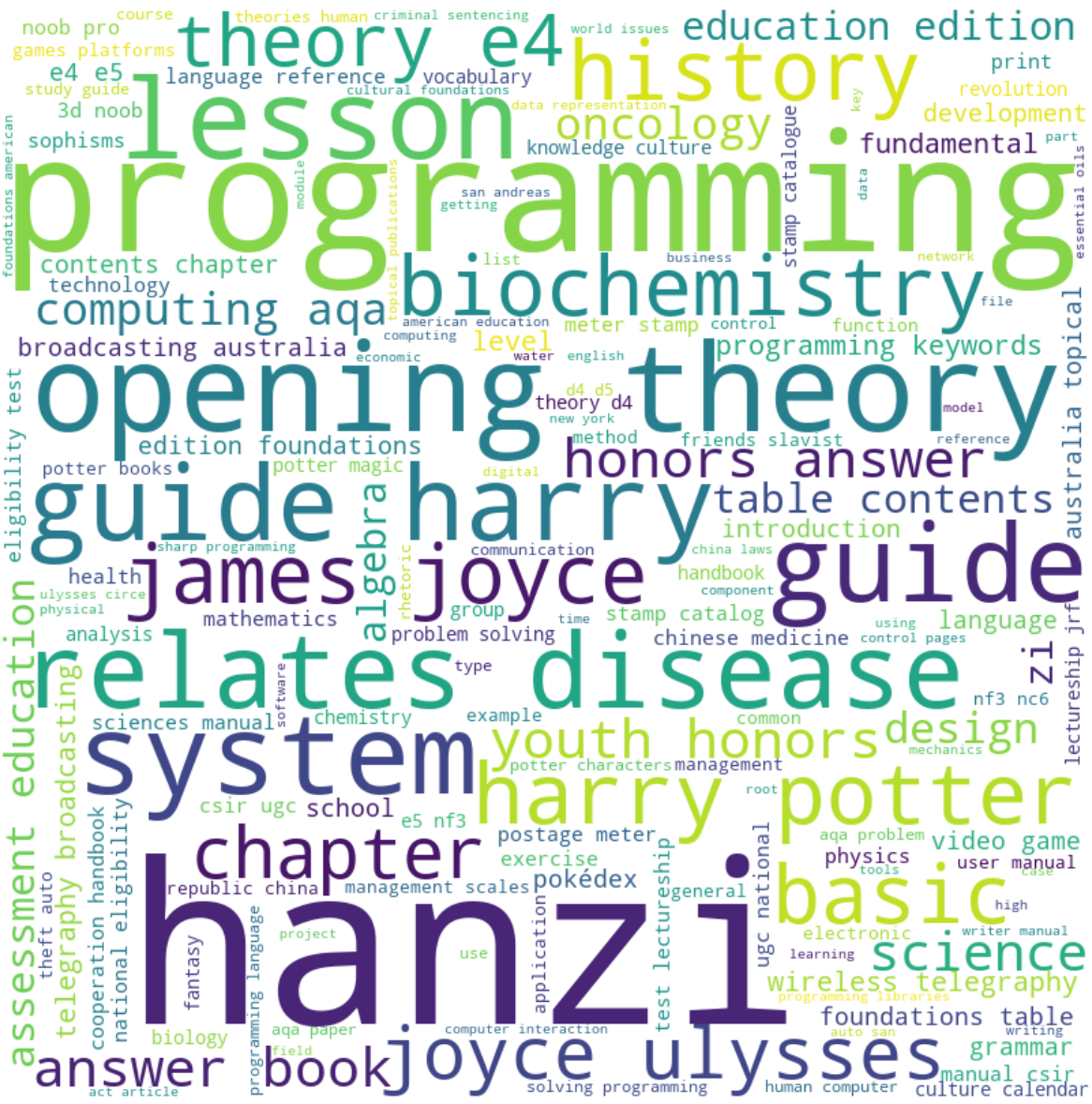
In [276]:
```
#turn dataframe into one large corpus (list)
corpus = []
for i in df['title_clean'][0:8000]:
    corpus.append(i)
```

WordCloud Visualization for Titles

In [287]:

```
#titles wordcloud
word_string = ''
for i in corpus:
    word_string += i
stopwords = set(STOPWORDS)
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(word_string)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
```



Creating tokens

Using keras.tokenizer I separated words into tokens. This transforms each word into a numerical value so I can create a padded sequence. I then feed the padded sequences into a LSTM model

In [13]:
```
tokenizer = Tokenizer()

def get_sequence_of_tokens(corpus):
    ## tokenization
    tokenizer.fit_on_texts(corpus)
    total_words = len(tokenizer.word_index) + 1

    ## convert corpus to sequence of tokens
    input_sequences = []
    for line in corpus:
        token_list = tokenizer.texts_to_sequences([line])[0]
        for i in range(1, len(token_list)):
            n_gram_sequence = token_list[:i+1]
            input_sequences.append(n_gram_sequence)
    return input_sequences, total_words

inp_sequences, total_words = get_sequence_of_tokens(corpus)
inp_sequences[:20]
```

Out[13]:
```
[[38, 55],
 [38, 55, 2148],
 [38, 55, 2148, 4140],
 [38, 55, 2148, 4140, 4141],
 [2149, 41],
 [2149, 41, 138],
 [4142, 705],
 [4142, 705, 450],
 [4142, 705, 450, 4143],
 [173, 73],
 [173, 73, 214],
 [173, 73, 214, 2804],
 [173, 73, 214, 2804, 4144],
 [173, 73, 214, 2804, 4144, 4145],
 [173, 73, 214, 2804, 4144, 4145, 950],
 [844, 27],
 [844, 27, 160],
 [844, 27, 160, 2805],
 [844, 27, 160, 2805, 4146],
 [844, 27, 160, 2805, 4146, 4147]]
```

In [14]:
```
def generate_padded_sequences(input_sequences):
    max_sequence_len = max([len(x) for x in input_sequences])
    input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_sequence_len, padding='pre'))

    predictors, label = input_sequences[:,:-1],input_sequences[:,-1]
    label = ku.to_categorical(label, num_classes=total_words)
    return predictors, label, max_sequence_len

predictors, label, max_sequence_len = generate_padded_sequences(inp_sequences)
```

LSTM Neural Network

I'm using a LSTM recurrent neural network to train my model. LSTM's are great for generative and predictive model.

My first layer is an embedded layer, this is to to transform the higher-dimentional data (arrays of text) into a lower-dementional format for the LSTM

I then have a hidden LSTM layer with the unit parameter at 100, followed by a dropout of 10%

Finally I have my output layer which is just a softmax dense layer

I compiled my model with the adam optimizer and a loss functino of categorical crossentropy

In [15]:

```
def create_model(max_sequence_len, total_words):
    input_len = max_sequence_len - 1
    model = Sequential()

    #embedding layer as input
    model.add(Embedding(total_words, 10, input_length=input_len))

    #1 hidden lstm layer with 10% dropout
    model.add(LSTM(100))
    model.add(Dropout(0.1))

    #output layer
    model.add(Dense(total_words, activation='softmax'))

    #compile the model
    model.compile(loss='categorical_crossentropy', optimizer='adam')

    return model

model = create_model(max_sequence_len, total_words)
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 44, 10)            93080

 lstm (LSTM)                 (None, 100)               44400

 dropout (Dropout)           (None, 100)               0

 dense (Dense)               (None, 9308)              940108

=================================================================
Total params: 1077588 (4.11 MB)
Trainable params: 1077588 (4.11 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [16]:
```
1 model.fit(predictors, label, epochs=100, verbose=3)
```

```
Epoch 1/100
Epoch 2/100
Epoch 3/100
Epoch 4/100
Epoch 5/100
Epoch 6/100
Epoch 7/100
Epoch 8/100
Epoch 9/100
Epoch 10/100
Epoch 11/100
Epoch 12/100
Epoch 13/100
Epoch 14/100
Epoch 15/100
Epoch 16/100
Epoch 17/100
Epoch 18/100
Epoch 19/100
Epoch 20/100
Epoch 21/100
Epoch 22/100
Epoch 23/100
Epoch 24/100
Epoch 25/100
Epoch 26/100
Epoch 27/100
Epoch 28/100
Epoch 29/100
Epoch 30/100
Epoch 31/100
Epoch 32/100
Epoch 33/100
Epoch 34/100
Epoch 35/100
Epoch 36/100
Epoch 37/100
Epoch 38/100
Epoch 39/100
Epoch 40/100
Epoch 41/100
Epoch 42/100
Epoch 43/100
Epoch 44/100
Epoch 45/100
Epoch 46/100
Epoch 47/100
Epoch 48/100
Epoch 49/100
Epoch 50/100
Epoch 51/100
Epoch 52/100
Epoch 53/100
Epoch 54/100
Epoch 55/100
Epoch 56/100
Epoch 57/100
Epoch 58/100
Epoch 59/100
Epoch 60/100
Epoch 61/100
Epoch 62/100
Epoch 63/100
Epoch 64/100
Epoch 65/100
Epoch 66/100
Epoch 67/100
Epoch 68/100
Epoch 69/100
Epoch 70/100
Epoch 71/100
Epoch 72/100
Epoch 73/100
Epoch 74/100
Epoch 75/100
Epoch 76/100
Epoch 77/100
Epoch 78/100
```

```
Epoch 79/100
Epoch 80/100
Epoch 81/100
Epoch 82/100
Epoch 83/100
Epoch 84/100
Epoch 85/100
Epoch 86/100
Epoch 87/100
Epoch 88/100
Epoch 89/100
Epoch 90/100
Epoch 91/100
Epoch 92/100
Epoch 93/100
Epoch 94/100
Epoch 95/100
Epoch 96/100
Epoch 97/100
Epoch 98/100
Epoch 99/100
Epoch 100/100
```

Out[16]:
```
<keras.src.callbacks.History at 0x2599bfbd410>
```

Save the model

In [17]:
```
model.save('titles2.keras')
```

Predicting Titles

In [23]:
```
def generate_title(input_text, next_words, model, max_sequence_len):
    for _ in range(next_words):
        #creates a list of tokens
        token_list = tokenizer.texts_to_sequences([input_text])[0]
        #creates a padded sequence from the list of tokens
        token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='pre')
        #predict the next word
        predicted = np.argmax(model.predict(token_list),axis=1)

        output_word = ""
        for word,index in tokenizer.word_index.items():
            if index == predicted:
                #add predicted text to output
                output_word = word
                break
        input_text += " "+output_word
    return 'Wikibooks: '+ input_text.title()
```

Some cherrypicked examples

In [25]:
```
generate_title('Science', 3, model, max_sequence_len)
```

```
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
```

Out[25]:
```
'Wikibooks: Science Elementary Teacher’S Guide'
```

In [26]:
```
generate_title('Chess', 4, model, max_sequence_len)
```

```
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
```

Out[26]:
```
'Wikibooks: Chess Opening Theory 1 E4'
```

In [27]:
```
1 generate_title('The', 4, model, max_sequence_len)
```

```
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 12ms/step
```

Out[27]: 'Wikibooks: The Programming Fundamentals Condition Examples'

In [28]:
```
1 generate_title('Python', 6, model, max_sequence_len)
```

```
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
```

Out[28]: 'Wikibooks: Python Programming Gui Programming Data Configuration Configuration'

In [33]:
```
1 generate_title('Bob', 3, model, max_sequence_len)
```

```
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
```

Out[33]: 'Wikibooks: Bob Shakespeare Works Comedies'

In [37]:
```
1 generate_title('A collection of', 4, model, max_sequence_len)
```

```
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 12ms/step
```

Out[37]: 'Wikibooks: A Collection Of Programming Code Statements Control'

Cleaning/Preprocessing book body

In [115]:
```
#an example of a text body
df['body_text'][11]
```

Out[115]: 'This Wikibooks page is a fact sheet and analysis on the article "Habitual physical activity in children and adolescents with cystic fibrosis" about how exercise is related to the disease Cystic Fibrosis.\n\nContents\n\n1 Background of this research\n2 Where is the research from\xa0?\n3 What kind of research was this?\n4 What did the research involve?\n\n4.1 Pulmonary Function testing\n4.2 Pros / Cons of this test\n\n\n5 What were the basic results?\n6 What conclusion can we take from this research\xa0?\n7 Practical Advice\n8 Further information/ Resources\n\n8.1 Cystic Fibrosis Australia\n8.2 Cystic Fibrosis\'s National Ambassador Nathan Charles\n\n\n9 References\n\n\n\nBackground of this research[edit\xa0| edit source]\nThe research was about the effects of taking part in exercise constantly or making it a habit in the population of children and teens that are severing from the genetic condition cystic Fibrosis.\n What is Cystic Fibrosis\nIt is a genetic condition, affecting lungs and digestion. Unfortunately, there is no cure. The condition Cystic Fibrosis (CF) is mostly inherited in the white population with 1 in every 3300 live births being diagnosed with the condition.[1]\n\nWhere is the research from\xa0?[edit\xa0| edit source]\nThis research was based in the American Children’s hospital Pittsburgh in the CF centre. Volunteers for this research included siblings, friends and hospital employee’s children who did not have the condition. Two authors of this research work within the department as paediatrics and others have conducted research regarding children with CF. This included David Michael Orenstein who has many publications on CF. These authors have also conducted other research on children with CF with methods of exercise that can help combat the condition.[2]\n\nWhat kind of research was this?[edit\xa0| edit source]\nThis was a meta-analysis form of research; even though this kind of research is time consuming the results are valid and reliable. Other studies that have been done have very similar results, regarding the effects of physical activity and the benefits it has on children and adolescents with CF. For an example, a study that was conducted in Austria compared the effects of physical activity versus chest physiotherapy which is popular within the CF community. [3] Two of the authors, David Michael and Patricia from the research article have conducted a study of “The prognostic value of exercise testing in patients with CF”.[2] Also, the Journal of Paediatric Pulmonology had similar conclusions that through exercise there is an improvement in oxygen consumption and physical self-efficiency and appearance in patients. As well as, lots of positive changes in living conditions of the patients.[4] Even though the research method used in these three studies differ, they all have very similar conclusions that exercise is beneficial for children with CF.\n\nWhat did the research involve?[edit\xa0| edit source]\n60 people in total 7–17 years of age[5]\n30 Patients with cystic fibrosis (18 male, 12 female)[5]\n30 people in the control not affected (17 male, 13 female)[5]\nThe participants completed a Questionnaire about their activity levels. Children 12 years and older completed it with no help or little assistant. Children 12 years and under did it with a parent or guardian. \nWhen getting tested the children did 2 types of tests, a Pulmonary Function test, and an Exercise Test. The level of aerobic fitness was tested by the participant completing a progressive exercise test on a stationary electronic bike (cycle ergometer) using the Godfrey protocol. Oxygen uptake was measured using a cart that you breathed into and then It analysed the breath content. This was recorded during the last 15 seconds of each stage exercise\n\nPulmonary Function testing[edit\xa0| edit source]\nPulmonary function was tested before exercising. Children who have CF had limited experience in doing these tests as they did not have regular exposure to the test due to the condition. A spirometry was used to measure pulmonary lung function capacity. The aim of the test is to measure how much and how quickly an individual is able to move air out of their lungs” ([6] this is done by breathing into a mouth piece connected to a device that records the air and it called a spirometer.\n\nPros / Cons of this test[edit\xa0| edit source]\nThis study was very good for testing but there were disadvantages on the younger population in the study due to being short as they were unable to reach the pedals. Another limitation of the study is that focus was only on the effects of aerobic training and did not take into account the benefit of anaerobic or resistance training can have on an individual. Also, the Australian Cystic Fibrosis Council suggest that core strength is also an important component of helping with the clearance of mucus for patients[1]\n\nWhat were the basic results?[edit\xa0| edit source]\nThe survival rate of living with Cystic Fibrosis is affected by the engagement of regulary physical Activity\nThe oxygen consumption improves with exercise.\nExercise helps with the removal of mucus\nChildren with Cystic Fibrosis participate in less vigorous physical exercise and activities when compared with children not affected by CF\nWhat conclusion can we take from this research\xa0?[edit\xa0| edit source]\nIn conclusion, this research demonstrates that exercise does have benefit\'s for children living with CF as it increases the survival rate and increase life expectancy. I believe one thing that is important when trying to help treat children with CF is to treat them normally and allowing them to engage in the activity as their peers are doing, within reason.\n\nPractical Advice[edit\xa0| edit source]\nBefore trying to treat CF with exercise consult Doctors about the type of exercise and don’t push yourself too hard. Build up the intensity.\n\nFurther information/ Resources[edit\xa0| edit source]\nCystic Fibrosis Australia[edit\xa0| edit source]\nCystic Fibrosis Australia even suggests that exercise is an important component of treating cystic fibrosis as it help with clearing the airways and building core strength.[1]\nWeb Page: http://www.cysticfibrosis.org.au\n\nCystic (http://www.cysticfibrosis.org.au\n\nCystic) Fibrosis\'s National Ambassador Nathan Charles[edit\xa0| edit source]\nCystic Fibrosis\'s National Ambassador Nathan Charles an elite rugby union player playing a contact sport while living with the condition cystic fibrosis. Shows that it is possible to stay fit and achieve great success with cystic fibrosis.[7]\nNathan Charles Web page http://nathancharles.com.au\nPlaying (http://nathancharles.com.au\nPlaying) Elite Rugby with CF: http://nathancharles.com.au/nutri-grain-unstoppable/\n\nReferences[edit\xa0| (http://nathancharles.com.au/nutri-grain-unstoppable/\n\nReferences[edit\xa0|) edit source]\n\n\n↑ a b c Cystic Fibrosis [Internet]. Cysticfibrosis.org.au. 2016 [cited 24 September 2016]. Available from: http://www.cysticfibrosis.org.au/all/learn/\n\n↑ (http://www.cysticfibrosis.org.au/all/learn/\n\n↑) a b Nixon P, Orenstein D, Kelsey S, Doershuk C. The prognostic value of exercise testing in patients with cystic fibrosis [Internet]. Saskatoon Public Library. 2010 [cited 15 September 2016]. Available from: http://saskatoonlibrary.ca/eds/item?dbid=edsgea&an=edsgcl.13305971\n\n↑ (http://saskatoonlibrary.ca/eds/item?dbid=edsgea&an=edsgcl.13305971\n\n↑) M. Orenstein D, A. Nixon P, A. Washburn , F. Kelsey S. Measuring Physical Activity in Children with Cystic Fibrosis: Comparison of Four Methods: Paediatric Exercise Science: Vol 5, No 2. Paediatric Exercise Science [Internet]. 2016 [cited 13 September 2016];5(2):125-133. Available from: http://journals.humankinetics.com/doi/pdf/10.1123/pes.5.2.125\n\n↑ (http://journals.humankinetics.com/doi/pdf/10.1123/pes.5.2.125\n\n↑) Gulmans V, de Meer K, Brackel H, Faber J, Berger R, Helders P. Outpatient exercise training in children with cystic fibrosis: Physiological effects, perceived competence, and acceptability. Pediatric Pulmonology [Internet]. 1999 [cited 15 September 2016];28(1):39-46. Available from: http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-0496(199907)28:1%3C39::AID-PPUL7%3E3.0.CO;2-8/abstract\n\n↑ (http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-0496(199907)28:1%3C39::AID-PPUL7%3E3.0.CO;2-8/abstract\n\n↑) a b c NIXON P, ORENSTEIN D, KELSEY S. Habitual physical activity in children and adolescents with CF. Medicine and Science in Sports and Exercise [Internet]. 2001 [cited 2 September 2016];33(1):30-35. Available from: http://zh9bf5sp6t.scholar.serialssolutions.com/?sid=google&auinit=PA&aulast=Nixon&atitle=Habitual+physical+activity+in+children+and+adolescents+**with**+cystic+fibrosis.&id=pmid:1119410\n\n↑ (http://zh9bf5sp6t.scholar.serialssolutions.com/?sid=google&auinit=PA&aulast=Nixon&atitle=Habitual+physical+activity+in+children+and+adolescents+with+cystic+fibrosis.&id=pmid:1119410\n\n↑) Lung Function Tests

[Internet]. WebMD. 2016 [cited 14 September 2016]. Available from: http://www.webmd.com/lung/lung-function-tests\n\n↑ (http://www.webmd.com/lung/lung-function-tests\n\n↑) Charles N. NATIONAL AMBASSADOR FOR CYSTIC FIBROSIS AUSTRALIA [Internet]. Nathan Charles. 2015 [cited 25 September 2016]. Available from: http://nathancharles.com.au/bio/' (http://nathancharles.com.au/bio/')

In [193]:
```
#creating my own punctuation list, i'm allowing the apostrophy to let contractions like don't to stay as one word
punctuation = string.punctuation.replace("'",'')
punctuation
```

Out[193]:
```
'!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [202]:
```
#function to clean text by removing punctuation, numbers, and irrelevant text.
nlp = spacy.load('en_core_web_sm')
def clean_body(txt, stem="None"):
    final_string = ""
    #remove uppercase
    txt = txt.lower()
    #remove line breaks
    txt = re.sub(r'\n', ' ', txt)
    #remove left in website text (only sometimes works?)
    txt = txt.replace('[edit\xa0| edit source]', ' ')
    #remove numbers
    txt = re.sub('[\d-]', ' ',txt)
    #remove unicode
    txt = re.sub(r'[^\x00-\x7F]', ' ', txt)
    #remove punctuation
    for i in punctuation:
        txt = txt.replace(i ,' ')
    #removes stray letters (leftover from punctuation being cleaned)
    #i put this multiple times because if two single letters were separated by a space it would only remove one o
    txt = re.sub("(^| ).( |$)", ' ', txt)
    txt = re.sub("(^| ).( |$)", ' ', txt)
    txt = re.sub("(^| ).( |$)", ' ', txt)
    #split txt
    txt = txt.split()

    #retrieve list of stopwords
    stop_words = nltk.corpus.stopwords.words("english")
    text_filtered = [word for word in txt if not word in stop_words]

    final_string = ' '.join(text_filtered)
    return final_string
```

In [203]:
```
df['body_clean'] = df['body_text'].apply(lambda x: clean_body(x))
```

In [204]:
```
#Look at same example cleaned
df['body_clean'][11]
```

Out[204]:
```
"wikibooks page fact sheet analysis article habitual physical activity children adolescents cystic fibrosis exercise
related disease cystic fibrosis contents background research research kind research research involve pulmonary funct
ion testing pros cons test basic results conclusion take research practical advice information resources cystic fibr
osis australia cystic fibrosis's national ambassador nathan charles references background research research effects
taking part exercise constantly making habit population children teens severing genetic condition cystic fibrosis cy
stic fibrosis genetic condition affecting lungs digestion unfortunately cure condition cystic fibrosis cf mostly inh
erited white population every live births diagnosed condition research research based american children hospital pit
tsburgh cf centre volunteers research included siblings friends hospital employee children condition two authors res
earch work within department paediatrics others conducted research regarding children cf included david michael oren
stein many publications cf authors also conducted research children cf methods exercise help combat condition kind r
esearch meta analysis form research even though kind research time consuming results valid reliable studies done sim
ilar results regarding effects physical activity benefits children adolescents cf example study conducted austria co
mpared effects physical activity versus chest physiotherapy popular within cf community two authors david michael pa
tricia research article conducted study prognostic value exercise testing patients cf also journal paediatric pulmon
ology similar conclusions exercise improvement oxygen consumption physical self efficiency appearance patients well
lots positive changes living conditions patients even though research method used three studies differ similar concl
usions exercise beneficial children cf research involve people total years age patients cystic fibrosis male female
people control affected male female participants completed questionnaire activity levels children years older comple
ted help little assistant children years parent guardian getting tested children types tests pulmonary function test
exercise test level aerobic fitness tested participant completing progressive exercise test stationary electronic bi
ke cycle ergometer using godfrey protocol oxygen uptake measured using cart breathed analysed breath content recorde
d last seconds stage exercise pulmonary function testing pulmonary function tested exercising children cf limited ex
perience tests regular exposure test due condition spirometry used measure pulmonary lung function capacity aim test
measure much quickly individual able move air lungs done breathing mouth piece connected device records air called s
pirometer pros cons test study good testing disadvantages younger population study due short unable reach pedals ano
ther limitation study focus effects aerobic training take account benefit anaerobic resistance training individual a
lso australian cystic fibrosis council suggest core strength also important component helping clearance mucus patien
ts basic results survival rate living cystic fibrosis affected engagement regulary physical activity oxygen consumpt
ion improves exercise exercise helps removal mucus children cystic fibrosis participate less vigorous physical exerc
ise activities compared children affected cf conclusion take research conclusion research demonstrates exercise bene
fit's children living cf increases survival rate increase life expectancy believe one thing important trying help tr
eat children cf treat normally allowing engage activity peers within reason practical advice trying treat cf exercis
e consult doctors type exercise push hard build intensity information resources cystic fibrosis australia cystic fib
rosis australia even suggests exercise important component treating cystic fibrosis help clearing airways building c
ore strength web page http www cysticfibrosis org au cystic fibrosis's national ambassador nathan charles cystic fib
rosis's national ambassador nathan charles elite rugby union player playing contact sport living condition cystic fi
brosis shows possible stay fit achieve great success cystic fibrosis nathan charles web page http nathancharles com
au playing elite rugby cf http nathancharles com au nutri grain unstoppable references cystic fibrosis internet cyst
icfibrosis org au cited september available http www cysticfibrosis org au learn nixon orenstein kelsey doershuk pro
gnostic value exercise testing patients cystic fibrosis internet saskatoon public library cited september available
http saskatoonlibrary ca eds item dbid edsgea edsgcl orenstein nixon washburn kelsey measuring physical activity chi
ldren cystic fibrosis comparison four methods paediatric exercise science vol paediatric exercise science internet c
ited september available http journals humankinetics com doi pdf pes gulmans de meer brackel faber berger helders ou
tpatient exercise training children cystic fibrosis physiological effects perceived competence acceptability pediatr
ic pulmonology internet cited september available http onlinelibrary wiley com doi sici aid ppul co abstract nixon o
renstein kelsey habitual physical activity children adolescents cf medicine science sports exercise internet cited s
eptember available http zh bf sp scholar serialssolutions com sid google auinit pa aulast nixon atitle habitual phys
ical activity children adolescents cystic fibrosis id pmid lung function tests internet webmd cited september availa
ble http www webmd com lung lung function tests charles national ambassador cystic fibrosis australia internet natha
n charles cited september available http nathancharles com au bio"
```

Create Tokens

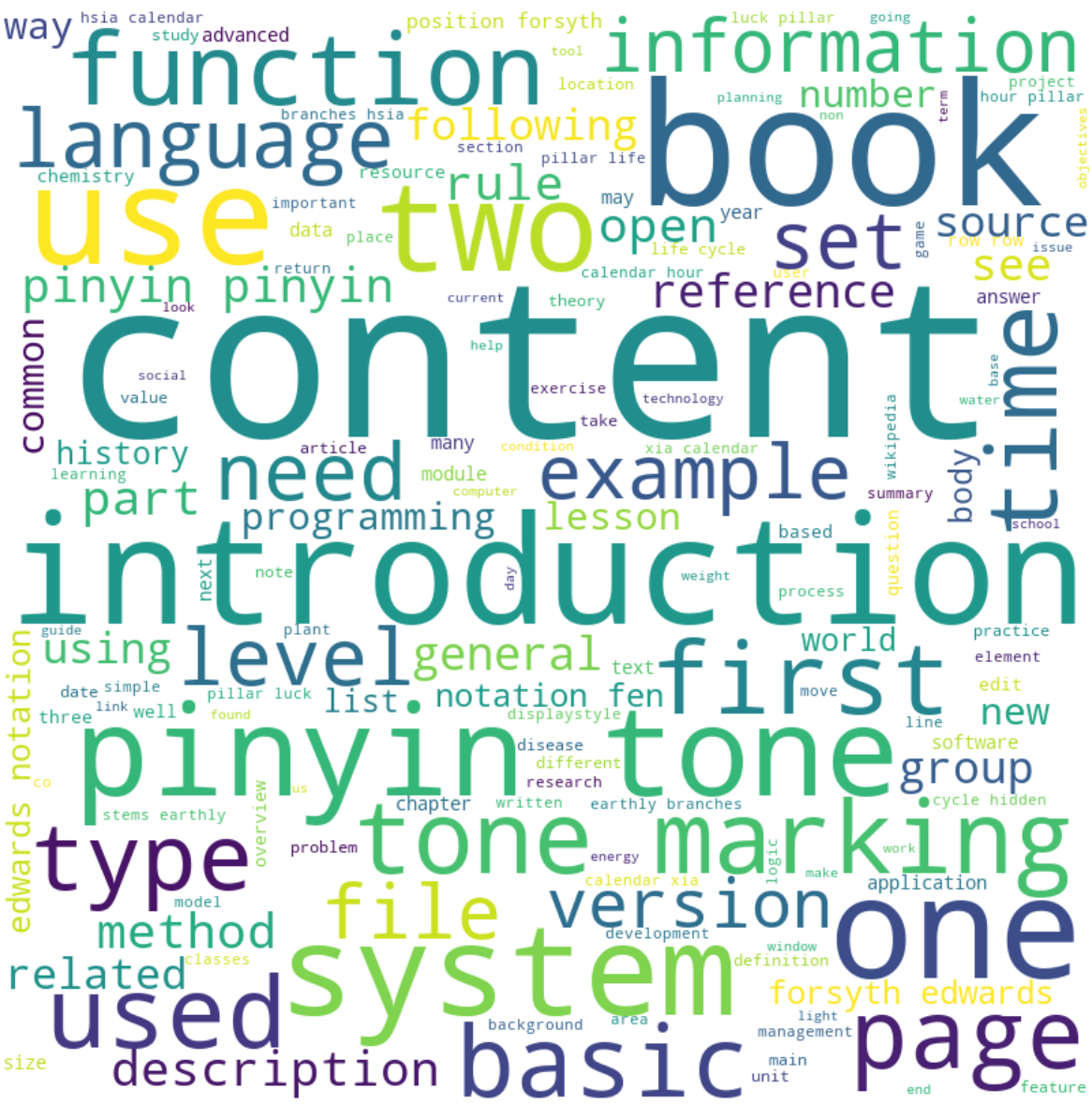
Tokenization, padded sequences, and model creation functions were already defined when working with the titles

In [288]:
```
corpus = []
#only take the first 800 results becasue of memory problems
for i in df['body_clean'][0:900]:
    wordlist = i.split()
    #only take 100 words because of memory problems
    corpus.append(i[10:110])
```

In [289]:

```
#body wordcloud
word_string = ''
for i in corpus:
    word_string += i
stopwords = set(STOPWORDS)
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(word_string)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
```



In [251]:
```
inp_sequences, total_words = get_sequence_of_tokens(corpus)
inp_sequences[:20]
```

Out[251]:
```
[[931, 1876],
 [931, 1876, 3211],
 [931, 1876, 3211, 1288],
 [931, 1876, 3211, 1288, 1675],
 [931, 1876, 3211, 1288, 1675, 1288],
 [931, 1876, 3211, 1288, 1675, 1288, 263],
 [931, 1876, 3211, 1288, 1675, 1288, 263, 12072],
 [931, 1876, 3211, 1288, 1675, 1288, 263, 12072, 6619],
 [931, 1876, 3211, 1288, 1675, 1288, 263, 12072, 6619, 154],
 [931, 1876, 3211, 1288, 1675, 1288, 263, 12072, 6619, 154, 43],
 [931, 1876, 3211, 1288, 1675, 1288, 263, 12072, 6619, 154, 43, 1675],
 [931, 1876, 3211, 1288, 1675, 1288, 263, 12072, 6619, 154, 43, 1675, 402],
 [33503, 1017],
 [33503, 1017, 12074],
 [33503, 1017, 12074, 4957],
 [33503, 1017, 12074, 4957, 12075],
 [33503, 1017, 12074, 4957, 12075, 64],
 [33503, 1017, 12074, 4957, 12075, 64, 5717],
 [33503, 1017, 12074, 4957, 12075, 64, 5717, 9982],
 [33503, 1017, 12074, 4957, 12075, 64, 5717, 9982, 197]]
```

Create padded seqence

In [252]:
```
predictors, label, max_sequence_len_body = generate_padded_sequences(inp_sequences)
```

Create LSTM model for body

In [253]:
```
model_body = create_model(max_sequence_len_body, total_words)
model_body.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_5 (Embedding)     (None, 22, 10)            654870

 lstm_5 (LSTM)               (None, 100)               44400

 dropout_5 (Dropout)         (None, 100)               0

 dense_5 (Dense)             (None, 65487)             6614187

=================================================================
Total params: 7313457 (27.90 MB)
Trainable params: 7313457 (27.90 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fit the model

In [254]:
```
model_body.fit(predictors, label, epochs=90, verbose=2)
```

```
342/342 - 23s - loss: 1.1226 - 23s/epoch - 67ms/step
Epoch 77/90
342/342 - 23s - loss: 1.1016 - 23s/epoch - 67ms/step
Epoch 78/90
342/342 - 23s - loss: 1.0728 - 23s/epoch - 67ms/step
Epoch 79/90
342/342 - 23s - loss: 1.0513 - 23s/epoch - 67ms/step
Epoch 80/90
342/342 - 23s - loss: 1.0444 - 23s/epoch - 66ms/step
Epoch 81/90
342/342 - 23s - loss: 1.0212 - 23s/epoch - 66ms/step
Epoch 82/90
342/342 - 22s - loss: 0.9988 - 22s/epoch - 65ms/step
Epoch 83/90
342/342 - 23s - loss: 0.9783 - 23s/epoch - 66ms/step
Epoch 84/90
342/342 - 23s - loss: 0.9585 - 23s/epoch - 67ms/step
Epoch 85/90
342/342 - 23s - loss: 0.9452 - 23s/epoch - 66ms/step
```

Save the model

In [255]:
```
model_body.save('body2.keras')
```

Generating a Wikibook with a small amount of text

In [256]:
```
#generates text from a title
def generate_body_from_title(input_text, next_words, model, max_sequence_len):
    words = []
    for _ in range(next_words):
        #creates a list of tokens from the input text
        token_list = tokenizer.texts_to_sequences([input_text])[0]
        token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='pre')
        predicted = np.argmax(model.predict(token_list),axis=1)

        output_word = ""
        for word,index in tokenizer.word_index.items():
            if index == predicted:
                output_word = word
                break
        input_text = input_text + " "+output_word
    return input_text
```

In [292]:
```
#generates a combination of a title and body text
def generate_book(input_text, title_length, text_length):
    title = generate_title(input_text, title_length ,model, max_sequence_len)
    body = generate_body_from_title(title[11:], text_length, model_body, max_sequence_len_body)
    print(title)
    print('')
    print(body)
```

In [293]:
```
generate_book('Midnight', 4,5)
```

```
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
Wikibooks: Midnight Revision Receiver Course 1

Midnight Revision Receiver Course 1 bone health financial output power
```

In [294]:
```
generate_book('Technology', 1,9)
```

```
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
Wikibooks: Technology 3

Technology 3 increases information editors briefly introduces reader output information energy
```

In [295]:
```
#for long texts it doesnt work so well, and sometimes there's other languages from books teaching another languag
generate_book('Fight', 1,60)
```

```
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
Wikibooks: Fight University

Fight University booklets print software edit information external assessment provides general version connection am
ount body us autopsy list body type na p expressions speed impul p displaystyl ma expressions corel ed whe fiends ze
brafish winged i cicero's cpa ylang cpa m tehdihng fla trav prani tomatoes klaptrap klobber trav meanings inheritanc
e timer st worsened statements conlang matlab viii fossil originally hambre quoi
```

In [296]:
```
#it gets off-topic quite fast
generate_book('Crime', 2,20)
```

```
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
Wikibooks: Crime File 1

Crime File 1 prehistoric times stone age bronze age imperial years qin southern dynas nearer s mass s group hand olo
ur ground host
```

In [299]:
```
generate_book('Silver', 2,9)
```

```
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 66ms/step
Wikibooks: Silver Body Comparison

Silver Body Comparison computer chemistry deals solely transition elements commomly called transition
```

In [307]:
```
#has a habbit of repeating words too
generate_book('Medical', 2,9)
```

```
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
Wikibooks: Medical Extent Int

Medical Extent Int herbs prescriptions source materia medica herb herb interactions sea
```

In [316]:
```
1 generate_book('Great', 1,9)
```

```
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
Wikibooks: Great Similar

Great Similar differentiation higher order derivatives second derivative second order derivativ
```

In [ ]:
```
1
```