1789-Washington.txt

word count: 1537 words

the 115 of 71 and 48 to 47 which 36 in 28

I 23 be 23 my 22 by 19 that 18 with 17

a 14 as 14 on 13 have 12 for 12 an 10 this 10 will 10

Average sentence length: 66.83

2009-Obama.txt

word count: 2700 words

the 126 and 105 of 82 to 66 our 58 we 50

that 48 a 47 is 36 us 23 not 22 in 22

are 22 this 20 will 19 can 19 but 17 have 16

for 15 on 15

Average sentence length: 24.55

2017-Trump.txt

word count: 1664 words

and 66 the 64 of 48 our 47 to 37 will 35 We 28 we 21 is 21 America 18 are 14

a 14 in 13 for 13 all 12 American 12

be 12 not 11 but 10 your 10

Average sentence length: 19.35

Word count: President Obama’s inaugural address in 2009 has 2700 words (including words, punctuation, numbers, etc.), which is 1000 more words than president Washington’s and president Trump’s inaugural address.

Twenty most frequent words: Across all three texts, the majority of 20 most frequent words consist of words of function classes. ‘the’, ‘and’, ‘of’ all are the top three common words. One interesting thing to point out, is that only president Trump’s inaugural address has ‘America’ and ‘American’ in the 20 most frequent words, which could be an indicator of the situation or the audience the president was addressing as.

Average sentence length: the average sentence length has steadily declined. Especially, there is a huge gap between the 1789’s and 2009’s inaugural address. This could be an indicator of how English evolved over 2 centuries, such that modern English’s average sentence length is much shorter than when US was just founded.

import nltk

#process\_wc\_top\_avglen(filename)

#This function takes in the filename, and returns the total word counts, NLTK

#frequency distribution, and the average sentence length as an array

def process\_wc\_top\_avglen(filename):

with open(filename) as f:

data = f.read()

sents = nltk.sent\_tokenize(data)

tokens = nltk.word\_tokenize(data)

words = [word for word in tokens if word.isalpha()]

fd = nltk.FreqDist(words)

total\_words = len(tokens)

avg\_sent\_len = total\_words / len(sents)

return (total\_words, fd, avg\_sent\_len)

#print\_wc\_top\_avglen(filename)

#This function takes in the filename, calls process\_wc\_top\_avglen(filename)

#and prints the filename, total wordcount, the freqdist, and average sentence

#length with some formatting

def print\_wc\_top\_avglen(filename):

result = process\_wc\_top\_avglen(filename)

avg\_len = round(result[2], 2)

print (filename)

print ('word count: ' + str(result[0]) + ' words')

print(result[1].most\_common(20))

print ("Average sentence length: " + str(avg\_len))

print ('')

def main():

print\_wc\_top\_avglen('1789-Washington.txt')

print\_wc\_top\_avglen('2009-Obama.txt')

print\_wc\_top\_avglen('2017-Trump.txt')

if \_\_name\_\_ == '\_\_main\_\_':

main()