Quiz 5: Spark APIs

Casey Owen

October 21, 2024

CS119 Big Data

Spring 2024

0.1 Setup

0.1.1 JDK Setup

[76]:
```
#Installing java 8
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
# -q, quiet level 2: no output except for errors
#> /dev/null on the end of any command where you want to redirect all the␣
↪stdout into nothingness
#Checking the installed Java version
!java -version
```

```
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu322.04)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu322.04, mixed mode,
sharing)
```

0.1.2 Install Spark

[77]:
```
!pip install -q findspark
!pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.10/dist-
packages (3.5.3)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-
packages (from pyspark) (0.10.9.7)
```

[78]:
```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
```

[79]:
```
from pyspark.sql.functions import lit
from pyspark.sql.functions import col
from pyspark.sql import functions as f
```

[80]:
```
sc = spark.sparkContext
```

0.2 Part I - Accumulators

0.2.1 Incorrect Code

[81]:
```
data = [1,2,3,4,5]
counter = 0
rdd = sc.parallelize(data)

# Wrong: Don't do this!!
def increment_counter(x):
    global counter
    counter += x

rdd.foreach(increment_counter)

print("Counter value: ", counter)
```

```
Counter value:  0
```

In the incorrect version, just the variable "counter" is used, which causes the behavior to be undefined. Each executor only sees a copy of the original counter, 0, and updates their individual copy without being able to see the copy on the driver. Thus, no updates happen, and 0 is returned.

0.2.2 Corrected Code:

[82]:
```
data = [1,2,3,4,5]
accum = sc.accumulator(0)
rdd = sc.parallelize(data)

def increment_counter(x):
    accum.add(x)

rdd.foreach(increment_counter)

print("Counter value: ", accum.value)
```

```
Counter value:  15
```

This version uses an accumulator, rather than a simple variable. The accumulator is supported by spark, and since it only supports being added to, it can be done efficiently in parallel. They are designed to allow safe updates to a variable in a distrributed environment, resulting in expected and consistent behaviour.

0.3 Part II - Airline Traffic

0.3.1 Question 1: Load and Clean Data

Get the data from the website directly using curl and unzip shell commands

```
[83]: june_url = r"https://www.bts.gov/sites/bts.dot.gov/files/docs/legacy/
        ↪additional-attachment-files/ONTIME.TD.202406.REL01.06AUG2024.zip"
      july_url = r"https://www.bts.gov/sites/bts.dot.gov/files/docs/legacy/
        ↪additional-attachment-files/ONTIME.TD.202407.REL01.03SEP2024.zip"

      !curl {june_url} > june_data.zip
      !curl {july_url} > july_data.zip

      !unzip -p july_data.zip > july_data.csv
      !unzip -p june_data.zip > june_data.csv

      # No longer need original archives
      !rm june_data.zip july_data.zip
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 21.8M  100 21.8M    0     0  36.1M      0 --:--:-- --:--:-- --:--:-- 36.1M
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 23.5M  100 23.5M    0     0  80.3M      0 --:--:-- --:--:-- --:--:-- 80.3M
```

Read dataframes from csv files

```
[84]: june_df = spark.read.csv('june_data.csv', sep="|", header=None)
      july_df = spark.read.csv('july_data.csv', sep="|", header=None)
      full_df = june_df.union(july_df)
```

Change column names to match provided schema

```
[85]: # All columns that we know what the schema is are labelled - others are unknown
      column_names = ["Carrier","Flight_Number", \
                      "Unknown1","Unknown2","Unknown3","Unknown4", \
                      "Departure_Airport", \
                      "Arrival_Airport", \
                      "Date_of_Flight_Operation_(Year/Month/Day)", \
                      "Day_of_Week_of_this_Flight_Operation_(Monday_=_1)", \
                      "Scheduled_Departure_Time_as_Shown_in_OAG", \
                      "Scheduled_Departure_Time_as_Shown_in_CRS", \
                     ␣
        ↪"Gate_Departure_Time_(Actual)_in_Local_Time","Scheduled_Arrival_Time_as_Shown_in_the_OAG",␣
        ↪\
                     ␣
        ↪"Scheduled_Arrival_Time_as_Shown_in_CRS","Gate_Arrival_Time_(Actual)_in_Local_Time",␣
        ↪\
                     ␣
        ↪"Difference_in_Minutes_Between_OAG_and_Scheduled_Departure_Time", \
                      "Difference_in_Minutes_Between_OAG_and_Scheduled_Arrival_Time",␣
        ↪\
                      "Scheduled_Elapsed_Time_Per_CRS_in_Minutes", \
                      "Actual_Gate_to_Gate_Time_in_Minutes", \
                      "Departure_Delay", \
                      "Arrival_Delay", \
                      "Elapsed_Time_Difference", \
                      "Wheels-Off_Time_(Actual)_in_Local_Time", \
                      "Wheels-On_Time_(Actual)_in_Local_Time", \
                      "Aircraft_Tail_Number", \
                      "Unknown5","Unknown6","Unknown7", \
                      "Cancellation Code", \
                      "Minutes_late_for_delay_code_E", \
                      "Minutes_late_for_delay_code_F", \
                      "Minutes_late_for_delay_code_G", \
                      "Minutes_late_for_delay_code_H", \
                      "Minutes_late_for_delay_code_I"]
      column_map = {}
      for i, column in enumerate(full_df.columns):
        if i >= len(column_names):
          break
        column_map[column] = column_names[i]
      full_df = full_df.withColumnsRenamed(column_map)
```

Drop columns where we do not know the schema/what the columns represent

[103]:
```
# Drop all columns not in schema - either marked with unknown, or never renamed
to_drop = []
for column in full_df.columns:
  if column.startswith('_c') or column.startswith('Unknown'):
    to_drop.append(column)

full_df = full_df.drop(*to_drop)
```

Load library of mappings from 2-Letter Airline IATA Codes to Airline names

[87]:
```
# Public online source for mapping of carrier codes to airline names
!curl https://gist.githubusercontent.com/AndreiCalazans/
 ↪390e82a1c3edff852137cb3da813eceb/raw/
 ↪1a1248f966b3f644f4eae057ad9b9b1b571c6aec/airlines.json > airline_codes.json
# Remove the second line from the file to clean - it is a duplicate
!sed '2d' airline_codes.json > airline_codes_cleaned.json

# Transpose horizontal dataframe to vertical by transposing, taking advantage␣
 ↪of Pandas .T - have to convert to pandas and back
# Needs to be vertical for joins
# This is a relatively small dataframe, so transpose is not too expensive
airline_codes_df = spark.read.json("airline_codes_cleaned.json",␣
 ↪multiLine=True) \
  .withColumn('IATA', lit('Airline')) \
  .to_pandas_on_spark() \
  .set_index('IATA') \
  .T \
  .reset_index() \
  .rename(columns={"index":"IATA"}) \
  .to_spark() \
  .withColumnsRenamed({"IATA":"Airline_Code","Airline":"Airline_Name"})


airline_codes_df.show()
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 26040  100 26040    0     0  73719      0 --:--:-- --:--:-- --:--:-- 73559
```

```
/usr/local/lib/python3.10/dist-packages/pyspark/sql/dataframe.py:5725:
FutureWarning: DataFrame.to_pandas_on_spark is deprecated. Use
DataFrame.pandas_api instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/pyspark/pandas/utils.py:1016:
PandasAPIOnSparkAdviceWarning: If `index_col` is not specified for `to_spark`,
the existing index is lost when converting to Spark DataFrame.
  warnings.warn(message, PandasAPIOnSparkAdviceWarning)
```

```
+------------+--------------------+
|Airline_Code|        Airline_Name|
+------------+--------------------+
|          0A|           Amber Air|
|          0B|            Blue Air|
|          0C|        IBL Aviation|
|          0D|      Darwin Airline|
|          0J|             Jetclub|
|          0V|Vietnam Air Servi…|
|          1A|     Amadeus IT Group|
|          1B|Abacus International|
|          1C|Electronic Data S…|
|          1D|Radixx Solutions …|
|          1E|Travelsky Technology|
|          1F|INFINI Travel Inf…|
|          1G|Galileo Internati…|
|          1H|         Siren-Travel|
|          1I|Sky Trek Internat…|
|          1K|                Sutra|
|          1L|Open Skies Consul…|
|          1M|JSC Transport Aut…|
|          1N|            Navitaire|
|          1P|            Worldspan|
+------------+--------------------+
only showing top 20 rows
```

Load library of mappings from 3-Letter Airport IATA Codes to Airport names

```
[88]: # Public online source for mapping of airport codes to airport names
!curl https://raw.githubusercontent.com/datasets/airport-codes/refs/heads/main/
  ↪data/airport-codes.csv > airport_codes.csv

airport_codes_df = spark.read.csv('airport_codes.csv', header=True) \
   .select('name', 'iata_code') \
   .where(col("iata_code").isNotNull()) \
   .withColumnsRenamed({"name": "Airport_Name", "iata_code": "Airport_Code"})

airport_codes_df.show()
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 8058k  100 8058k    0     0  32.0M      0 --:--:-- --:--:-- --:--:-- 32.1M
+--------------------+------------+
|        Airport_Name|Airport_Code|
+--------------------+------------+
|      Utirik Airport|         UTK|
|Ocean Reef Club A…|         OCA|
|       Cuddihy Field|         CUX|
|Crested Butte Air…|         CSE|
|    Columbus Airport|         CUS|
|   LBJ Ranch Airport|         JCY|
|Loring Seaplane Base|         WLR|
| Nunapitchuk Airport|         NUP|
|Port Alice Seapla…|         PTC|
|     Icy Bay Airport|         ICY|
|Port Protection S…|         PPV|
|Kalakaket Creek A…|         KKK|
|Dunsmuir Muni-Mot…|         MHS|
|Chase Field Indus…|         NIR|
|Grand Canyon Bar …|         GCT|
|Ellamar Seaplane …|         ELW|
|Lime Village Airport|         LVD|
|   Hog River Airport|         HGZ|
|      Ed-Air Airport|         OTN|
|      Telida Airport|         TLF|
+--------------------+------------+
only showing top 20 rows
```

0.3.2 Question 2: What US Airline Has the Least Delays

```
[89]: # Consider "delay" to be the average of arrival and departure delays
q2_df = full_df \
  .withColumn('Mean_Delay', (full_df.Departure_Delay + full_df.Arrival_Delay)/
 ↪2) \
  .groupBy("Carrier").avg("Mean_Delay") \
  .join(airline_codes_df,full_df.Carrier == airline_codes_df.Airline_Code,␣
 ↪"inner") \
  .withColumn("Average_Delay_Minutes", f.round("avg(Mean_Delay)",2)) \
  .drop('Airline_Code', 'Carrier', 'avg(Mean_Delay)') \
  .sort(col("Average_Delay_Minutes").asc()) \
  .show(truncate=False)
```

```
+---------------------+---------------------+
|Airline_Name         |Average_Delay_Minutes|
+---------------------+---------------------+
|Hawaiian Airlines    |5.33                 |
|Alaska Airlines, Inc.|7.2                  |
|Southwest Airlines   |14.09                |
|United Airlines      |15.95                |
|Delta Air Lines      |16.12                |
|Allegiant Air        |17.96                |
|Spirit Airlines      |19.38                |
|JetBlue Airways      |20.17                |
|American Airlines    |22.06                |
|Frontier Airlines    |23.66                |
+---------------------+---------------------+
```

Hawaiian Airlines has the lowest average delay

0.3.3 Question 3: What Departure Time of Day is Best to Avoid

```
[90]: # Again using delay as the average of arrival and departure
q3_df = full_df \
  .withColumn('Time_of_Day',
             f.when(col('Scheduled_Departure_Time_as_Shown_in_OAG').
 ↪between(600, 959),'morning(6AM-10AM)') \
             .when(col('Scheduled_Departure_Time_as_Shown_in_OAG').
 ↪between(1000, 1359),'midday(10AM-2PM)') \
             .when(col('Scheduled_Departure_Time_as_Shown_in_OAG').
 ↪between(1400, 1759),'afternoon(2PM-6PM)') \
             .when(col('Scheduled_Departure_Time_as_Shown_in_OAG').
 ↪between(1800, 2159),'evening(6PM-10PM)') \
             .otherwise('night(10PM-6AM)')
             ) \
  .withColumn('Mean_Delay', (full_df.Departure_Delay + full_df.Arrival_Delay)/
↪2) \
  .groupBy("Time_of_Day").avg("Mean_Delay") \
  .withColumn("Average_Delay_Minutes", f.round("avg(Mean_Delay)",2)) \
  .drop('avg(Mean_Delay)') \
  .sort(col("Average_Delay_Minutes").asc()) \
  .show(truncate=False)
```

```
+------------------+---------------------+
|Time_of_Day       |Average_Delay_Minutes|
+------------------+---------------------+
|morning(6AM-10AM) |5.98                 |
|midday(10AM-2PM)  |12.41                |
|night(10PM-6AM)   |16.33                |
|afternoon(2PM-6PM)|23.45                |
|evening(6PM-10PM) |29.93                |
+------------------+---------------------+
```

Morning (6AM-10AM) has the lowest average delay

0.3.4 Question 4: What Airports Have the Most Flight Delays

[91]:
```
from pyspark.sql.types import IntegerType

# Need to seperately track departure delays when a given airport was the␣
↪departure airport, and arrival delays when a given airport was the arrival␣
↪airport
q4_departure_delays_df = full_df \
  .withColumn('Departure_Delay_int', full_df['Departure_Delay'].
↪cast(IntegerType())) \
  .groupBy("Departure_Airport").avg("Departure_Delay_int") \
  .withColumnsRenamed({"avg(Departure_Delay_int)":"Average_Departure_Delay"}) \

q4_arrival_delays_df = full_df \
  .withColumn('Arrival_Delay_int', full_df['Arrival_Delay'].
↪cast(IntegerType())) \
  .groupBy("Arrival_Airport").avg("Arrival_Delay_int") \
  .withColumnsRenamed({"avg(Arrival_Delay_int)":"Average_Arrival_Delay"}) \

# Now consider "delay" to be the SUM of arrival and departure delays
q4_df = q4_departure_delays_df \
  .join(q4_arrival_delays_df, q4_departure_delays_df.Departure_Airport ==␣
↪q4_arrival_delays_df.Arrival_Airport) \
  .withColumn('Average_Total_Delay', q4_departure_delays_df.
↪Average_Departure_Delay + q4_arrival_delays_df.Average_Arrival_Delay) \
  .withColumn('Average_Total_Delay_Minutes', f.round("Average_Total_Delay",2)) \
  .join(airport_codes_df, q4_departure_delays_df.Departure_Airport ==␣
↪airport_codes_df.Airport_Code) \
  .drop('Arrival_Airport', 'Departure_Airport', 'Average_Departure_Delay',␣
↪'Average_Arrival_Delay', 'Average_Total_Delay', 'Airport_Code') \
  .sort(col("Average_Total_Delay_Minutes").desc()) \
  .show(n=20, truncate=False)
```

```
+---------------------------+---------------------------------------------------
----+
|Average_Total_Delay_Minutes|Airport_Name
|
+---------------------------+---------------------------------------------------
----+
|85.43                      |Hagerstown Regional Richard A Henson Field
|
|83.2                       |Lea County Regional Airport
|
|81.21                      |Elko Regional Airport
|
|78.11                      |Mason City Municipal Airport
|
|75.19                      |Sioux Gateway Airport / Brigadier General Bud Day
Field|
|74.75                      |Concord-Padgett Regional Airport
|
|72.07                      |Bemidji Regional Airport
|
|71.52                      |Eastern Sierra Regional Airport
|
|68.39                      |Stockton Metropolitan Airport
|
|66.06                      |Henry E Rohlsen Airport
|
|62.03                      |Roswell Air Center Airport
|
|61.38                      |Yeager Airport
|
|59.55                      |Texarkana Regional Airport (Webb Field)
|
|58.54                      |Elmira Corning Regional Airport
|
|58.5                       |Pago Pago International Airport
|
|57.44                      |Dothan Regional Airport
|
|57.2                       |Fort Dodge Regional Airport
|
|55.53                      |Rafael Hernández International Airport
|
|54.69                      |Provo-Utah Lake International Airport
|
|54.48                      |La Crosse Regional Airport
|
+---------------------------+-------------------------------------------------
----+
only showing top 20 rows
```

Hagerstown Regional Richard A Henson Field had the highest total delay average

0.3.5 Question 5: What Airports are the Top 5 Busiest Airports in the US

[92]:
```
# Seperately count departures when a given airport was the departure airport,␣
↪and arrivals when a given airport was the arrival airport
q5_departure_df = full_df \
  .groupBy('Departure_Airport').count() \
  .withColumnsRenamed({"count":"Total_Departures"})

q5_arrival_df = full_df \
  .groupBy('Arrival_Airport').count() \
  .withColumnsRenamed({"count":"Total_Arrivals"})

q5_df = q5_departure_df \
  .join(q5_arrival_df, q5_departure_df.Departure_Airport == q5_arrival_df.
↪Arrival_Airport) \
  .withColumn('Total_Arrivals_And_Departures', q5_departure_df.Total_Departures␣
↪+ q5_arrival_df.Total_Arrivals) \
  .join(airport_codes_df, q5_departure_df.Departure_Airport == airport_codes_df.
↪Airport_Code) \
  .drop('Arrival_Airport', 'Departure_Airport', 'Total_Departures',␣
↪'Total_Arrivals', 'Airport_Code') \
  .sort(col("Total_Arrivals_And_Departures").desc()) \
  .show(n=5, truncate=False)
```

```
+-----------------------------+------------------------------------------------+
|Total_Arrivals_And_Departures|Airport_Name                                    |
+-----------------------------+------------------------------------------------+
|119128                       |Hartsfield Jackson Atlanta International Airport|
|114319                       |Dallas Fort Worth International Airport         |
|111962                       |Denver International Airport                    |
|111469                       |Chicago O'Hare International Airport            |
|87671                        |Charlotte Douglas International Airport         |
+-----------------------------+------------------------------------------------+
only showing top 5 rows
```

0.4 Part III - ShortStoryJam

Load Data

[93]:
```
!rm -rf big-data-repo/
!git clone https://github.com/singhj/big-data-repo.git
```

```
Cloning into 'big-data-repo'…
remote: Enumerating objects: 548, done.
remote: Counting objects: 100% (125/125), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 548 (delta 117), reused 109 (delta 109), pack-reused 423 (from
1)
Receiving objects: 100% (548/548), 56.47 MiB | 22.55 MiB/s, done.
Resolving deltas: 100% (295/295), done.
```

0.4.1 Question 1: Clean Text and Remove Stopwords

[94]:
```
import requests
import re
import string

stopwords_list = requests.get("https://gist.githubusercontent.com/rg089/
↪35e00abf8941d72d419224cfd5b5925d/raw/
↪12d899b70156fd0041fa9778d657330b024b959c/stopwords.txt").content
stopwords = list(set(stopwords_list.decode().splitlines()))

def remove_stopwords(words):
  list_ = re.sub(r"[^a-zA-Z0-9]", " ", words.lower()).split()
  return [itm for itm in list_ if itm not in stopwords]

def clean_text(text):
  text = text.lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), ' ', text)
  text = re.sub('[\d\n]', ' ', text)
  return ' '.join(remove_stopwords(text))
```

Create function "clean_book" that cleans the book of a provided name and returns the cleaned text as a single string

[95]:
```
def clean_book(book_name):
  folder_path = 'big-data-repo/text-proc/poe-stories/'
  with open(folder_path + book_name) as f:
    cleaned = clean_text(f.read())
  f.close()
  return cleaned

# Testing by outputting first 100 characters of cleaned string
```

```
print(clean_book('A_DESCENT_INTO_THE_MAELSTROM')[:100])
```

```
ways god nature providence ways models frame commensurate vastness profundity
unsearchableness works
```

0.4.2 Question 2: Use NLTK to decompose the first story A_DESCENT_INTO_THE_MAELSTROM

[96]:
```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

[96]:
```
True
```

[97]:
```
paragraph = clean_book('A_DESCENT_INTO_THE_MAELSTROM')

def get_pos_tuples(text):
  sent_text = nltk.sent_tokenize(text) # this gives us a list of sentences
  # now loop over each sentence and tokenize it separately
  return [nltk.pos_tag(nltk.word_tokenize(sent)) for sent in sent_text][0]

# Printing first 5 as a test
print(get_pos_tuples(paragraph)[:5])
```

```
[('ways', 'NNS'), ('god', 'VBP'), ('nature', 'JJ'), ('providence', 'NN'),
('ways', 'NNS')]
```

0.4.3 Question 3: Tag non-stopwords as parts of speech using Penn POS Tags

[98]:
```
from nltk.data import load
nltk.download('tagsets')
taginfo = load('help/tagsets/upenn_tagset.pickle')
```

```
[nltk_data] Downloading package tagsets to /root/nltk_data…
[nltk_data]   Package tagsets is already up-to-date!
```

[99]:
```
from functools import reduce

def append_if_match_tag(tag_list:list, pos_tuple, tag):
  word, pos = pos_tuple
  if pos == tag:
    return tag_list + [word]
  else:
    return tag_list

def get_tag_words(tag, tagged_tuples:list[tuple]):
  # Returns list of words with that tag
  return (tag, reduce(lambda tag_list, pos_tuple: append_if_match_tag(tag_list,␣
↪pos_tuple, tag), tagged_tuples, []))

def get_tag_dict(text, taginfo):
  '''
  Gets the full tag dictionary of a given piece of clean text
  '''
  possible_tags = taginfo.keys()
  pos_tuples = get_pos_tuples(text)
  return dict(list(map(lambda tag: get_tag_words(tag, pos_tuples),␣
↪possible_tags)))

tag_dict = get_tag_dict(paragraph, taginfo)

# Printing entire dict as per assignment spec
print(tag_dict)
```

```
{'LS': [], 'TO': [], 'VBN': ['endured', 'beheld', 'reared', 'called',
'ascended', 'set', 'acquired', 'lashed', 'phrensied', 'assumed', 'round',
'heard', 'received', 'gazed', 'confessed', 'agreed', 'rigged', 'set', 'thought',
'driven', 'drifted', 'rendered', 'weighed', 'set', 'headed', 'batten', 'buried',
'escaped', 'held', 'doomed', 'blazed', 'trimmed', 'recognised', 'possessed',
'headed', 'condemned', 'careered', 'approached', 'secured', 'expected',
'ceased', 'confused', 'revolved', 'met', 'broken', 'floated', 'deceived', 'set',
'called', 'absorbed', 'drawn', 'absorbed', 'explained', 'moved', 'cut',
'thought', 'fastened', 'secured', 'precipitated', 'plunged', 'picked'], "''":
[], 'WP': [], 'UH': [], 'VBG': ['falling', 'shining', 'particularizing',
'beetling', 'howling', 'shrieking', 'blowing', 'offing', 'dashing',
'increasing', 'moaning', 'chopping', 'changing', 'heaving', 'boiling',
'hissing', 'gyrating', 'whirling', 'plunging', 'spreading', 'entering',
'gleaming', 'shining', 'speeding', 'swaying', 'sweltering', 'sending',
'appalling', 'bewildering', 'conveying', 'splitting', 'returning', 'guarding',
'attempting', 'howling', 'smiling', 'coming', 'rising', 'falling',
'penetrating', 'issuing', 'mentioning', 'standing', 'answering', 'coming',
'starving', 'starting', 'fishing', 'dreaming', 'knowing', 'proposing',
'drifting', 'driving', 'describing', 'chopping', 'ascertaining', 'grasping',
'ring', 'keeping', 'coming', 'driving', 'frothing', 'riding', 'falling',
'roaring', 'letting', 'writhing', 'approaching', 'boasting', 'flying',
'floating', 'ring', 'holding', 'ring', 'raving', 'swaying', 'sickening',
'falling', 'hanging', 'gleaming', 'lying', 'observing', 'maintaining',
'tottering', 'clashing', 'dizzying', 'building', 'speculating', 'exciting',
'entering', 'undergoing', 'floating', 'swimming', 'startling', 'enforcing',
'rendering', 'floating', 'ring', 'quitting', 'bearing', 'setting'], 'JJ':
['nature', 'commensurate', 'unsearchableness', 'speak', 'mortal', 'suppose',
'single', 'jetty', 'black', 'white', 'unstring', 'tremble', 'cliff', 'thrown',
'extreme', 'unobstructed', 'black', 'sixteen', 'half', 'perilous', 'vain',
'danger', 'long', 'sufficient', 'brought', 'close', 'norwegian', 'eighth',
'degree', 'great', 'dreary', 'giddy', 'wide', 'ocean', 'inky', 'nubian',
'desolate', 'human', 'outstretched', 'black', 'cliff', 'high', 'white',
'opposite', 'visible', 'small', 'discernible', 'arose', 'craggy', 'ocean',
'unusual', 'strong', 'landward', 'double', 'hull', 'short', 'angry', 'moskoe',
'ambaaren', 'suarven', 'stockholm', 'true', 'understand', 'hear', 'interior',
'lofoden', 'burst', 'aware', 'loud', 'vast', 'american', 'prairie', 'seamen',
'ocean', 'current', 'gazed', 'current', 'monstrous', 'headlong', 'ungovernable',
'main', 'uproar', 'vast', 'scarred', 'gigantic', 'innumerable', 'eastward',
'precipitous', 'radical', 'general', 'smooth', 'prodigious', 'apparent',
'great', 'form', 'distinct', 'broad', 'terrific', 'smooth', 'black', 'shriek',
'niagara', 'agony', 'threw', 'scant', 'nervous', 'great', 'whirlpool', 'moskoe',
'str', 'ordinary', 'prepared', 'circumstantial', 'wild', 'feeble', 'lofoden',
'afford', 'convenient', 'flood', 'lofoden', 'boisterous', 'impetuous',
'dreadful', 'depth', 'carried', 'thrown', 'ebb', 'weather', 'boisterous',
'dangerous', 'mile', 'impossible', 'fruitless', 'bear', 'swim', 'lofoden',
'stream', 'large', 'current', 'torn', 'consist', 'fro', 'flux', 'high', 'low',
'noise', 'depth', 'shore', 'lofoden', 'sidelong', 'difficult', 'evident',
'attraction', 'phenomenon', 'plausible', 'unsatisfactory', 'flux', 'natural',
'prodigious', 'remote', 'idle', 'hear', 'subject', 'conclusive',
'unintelligible', 'good', 'deaden', 'proceeded', 'vurrgh', 'violent', 'good',
'proper', 'attempt', 'lofoden', 'regular', 'usual', 'great', 'preferred',
'single', 'desperate', 'main', 'str', 'otterholm', 'remain', 'slack', 'steady',
'fail', 'mis', 'stay', 'dead', 'rare', 'arrival', 'boisterous', 'round',
'fouled', 'innumerable', 'lee', 'good', 'twentieth', 'bad', 'good', 'str',
'minute', 'strong', 'current', 'unmanageable', 'eighteen', 'great', 'young',
'horrible', 'tenth', 'blew', 'terrible', 'late', 'steady', 'follow', 'smack',
'fine', 'plenty', 'fresh', 'starboard', 'great', 'unusual', 'feel', 'uneasy',
'astern', 'singular', 'amazing', 'dead', 'long', 'smack', 'attempt',
'experienced', 'feather', 'complete', 'small', 'cross', 'seas', 'foresail',
'flat', 'narrow', 'gunwale', 'bolt', 'mere', 'undoubtedly', 'hold', 'clear',
'rid', 'collect', 'grasp', 'mouth', 'close', 'str', 'violent', 'fit', 'meant',
'wished', 'whirl', 'perceive', 'str', 'whirl', 'hope', 'great', 'fool', 'gun',
'ship', 'seas', 'lay', 'flat', 'absolute', 'singular', 'black', 'circular',
'clear', 'clear', 'deep', 'blue', 'wear', 'lit', 'god', 'light', 'understand',
'hear', 'single', 'shook', 'pale', 'hideous', 'dragged', 'ocean', 'str', 'deep',
'strong', 'gale', 'large', 'strange', 'gigantic', 'sky', 'high', 'sick',
'lofty', 'quick', 'sufficient', 'exact', 'str', 'whirlpool', 'dead', 'str',
'foam', 'sharp', 'larboard', 'shot', 'noise', 'sound', 'imagine', 'whirl',
'thought', 'amazing', 'borne', 'bubble', 'starboard', 'ocean', 'stood', 'huge',
'strange', 'rid', 'great', 'suppose', 'strung', 'reflect', 'magnificent',
'foolish', 'individual', 'wonderful', 'shame', 'depths', 'principal',
'singular', 'general', 'high', 'black', 'mountainous', 'heavy', 'gale', 'great',
'rid', 'petty', 'forbidden', 'uncertain', 'impossible', 'middle', 'horrible',
'stern', 'small', 'coop', 'counter', 'gale', 'large', 'afford', 'madman',
'maniac', 'bolt', 'astern', 'great', 'fro', 'immense', 'lurch', 'headlong',
'hurried', 'sweep', 'open', 'instant', 'lived', 'foam', 'magic', 'interior',
'vast', 'prodigious', 'smooth', 'ebony', 'circular', 'flood', 'golden', 'black',
'inmost', 'general', 'downward', 'unobstructed', 'surface', 'parallel',
'footing', 'dead', 'thick', 'hung', 'narrow', 'great', 'dare', 'foam', 'great',
'descent', 'uniform', 'complete', 'downward', 'perceptible', 'wide', 'liquid',
'visible', 'large', 'timber', 'unnatural', 'original', 'grow', 'dreadful',
'watch', 'strange', 'numerous', 'delirious', 'relative', 'fir', 'tree', 'awful',
'invariable', 'tremble', 'terror', 'great', 'buoyant', 'coast', 'lofoden',
'thrown', 'extraordinary', 'stuck', 'disfigured', 'entered', 'late', 'level',
'ocean', 'general', 'descent', 'equal', 'extent', 'spherical', 'sphere',
'equal', 'cylindrical', 'escape', 'subject', 'forgotten', 'natural', 'bulky',
'great', 'anxious', 'vessel', 'high', 'original', 'resolved', 'loose',
'impossible', 'cask', 'tale', 'bring', 'vast', 'wild', 'rapid', 'foam', 'great',
'vast', 'steep', 'violent', 'gulf', 'uprise', 'clear', 'surface', 'ocean',
'mountainous', 'coast', 'exhausted', 'speechless', 'daily', 'raven', 'black',
'white', 'told', 'merry'], 'VBZ': ['works', 'hairs', 'assumes', 'passages',
'decreases', 'runs', 'leagues', 'whales', 'rocks', 'precipitates', 'rises',
'length', 'boys', 'dark', 'heavens', 'appears', 'surrounds', 'walls', 'ends',
'staves', 'takes', 'masses'], '--': [], 'VBP': ['god', 'frame', 'terror',
'brink', 'mountain', 'guide', 'sea', 'land', 'remote', 'swell', 'island',
'vurrgh', 'sandflesen', 'helseggen', 'sea', 'burst', 'scene', 'foam', 'length',
'vortex', 'ramus', 'beholder', 'calmest', 'roar', 'extent', 'rocks', 'turn',
'disengage', 'firs', 'coast', 'channel', 'whales', 'disappear', 'account',
'ferroe', 'cataract', 'encyclop', 'comprehend', 'creep', 'burthen', 'moskoe',
'sea', 'courage', 'yield', 'slack', 'violent', 'sweeps', 'forget', 'apprehend',
'point', 'puff', 'head', 'boat', 'stupor', 'save', 'wind', 'speak', 'meant',
'flung', 'wave', 'expect', 'turn', 'shriek', 'account', 'gulf', 'shore',
'doubt', 'occupy', 'hold', 'whirl', 'awe', 'mistaken', 'moon', 'abyss', 'moon',
'account', 'pathway', 'funnel', 'mist', 'foam', 'hope', 'rapid', 'whirlpool',
'understand', 'whirl', 'froth', 'lofoden', 'hurricane', 'boat', 'expect',
'lofoden'], 'NN': ['providence', 'vastness', 'profundity', 'democritus',
'joseph', 'glanville', 'summit', 'crag', 'man', 'length', 'route', 'event',
'man', 'man', 'body', 'soul', 'man', 'day', 'change', 'weaken', 'exertion',
'giddy', 'cliff', 'edge', 'rest', 'portion', 'body', 'tenure', 'slippery',
'edge', 'cliff', 'sheer', 'precipice', 'rock', 'dozen', 'truth', 'position',
'companion', 'ground', 'clung', 'shrubs', 'glance', 'sky', 'idea', 'fury',
'reason', 'courage', 'sit', 'distance', 'view', 'scene', 'event', 'story',
'spot', 'eye', 'manner', 'coast', 'latitude', 'province', 'nordland',
'district', 'lofoden', 'mountain', 'sit', 'helseggen', 'cloudy', 'hold',
'grass', 'feel', 'belt', 'vapor', 'beneath', 'sea', 'expanse', 'hue', 'bring',
'mind', 'geographer', 'account', 'mare', 'tenebrarum', 'panorama',
'imagination', 'conceive', 'eye', 'reach', 'character', 'gloom', 'surf',
'promontory', 'apex', 'distance', 'bleak', 'island', 'position', 'wilderness',
'surge', 'size', 'cluster', 'dark', 'appearance', 'space', 'distant', 'island',
'time', 'gale', 'brig', 'lay', 'reefed', 'trysail', 'sight', 'regular', 'quick',
'cross', 'water', 'direction', 'wind', 'foam', 'vicinity', 'distance', 'man',
'mile', 'islesen', 'hotholm', 'buckholm', 'moskoe', 'vurrgh', 'change', 'water',
'caught', 'glimpse', 'sea', 'summit', 'man', 'sound', 'herd', 'moment', 'term',
'character', 'beneath', 'velocity', 'moment', 'speed', 'impetuosity', 'fury',
'moskoe', 'coast', 'bed', 'conflicting', 'convulsion', 'rapidity', 'water',
'alteration', 'surface', 'distance', 'combination', 'gyratory', 'motion',
'germ', 'vast', 'definite', 'existence', 'circle', 'mile', 'diameter', 'edge',
'whirl', 'belt', 'spray', 'particle', 'mouth', 'funnel', 'interior', 'eye',
'fathom', 'jet', 'wall', 'water', 'horizon', 'angle', 'round', 'motion',
'voice', 'half', 'half', 'roar', 'mighty', 'cataract', 'mountain', 'base',
'rock', 'face', 'clung', 'herbage', 'agitation', 'length', 'man', 'maelstr',
'island', 'moskoe', 'midway', 'impart', 'conception', 'magnificence', 'horror',
'scene', 'sense', 'point', 'view', 'writer', 'question', 'time', 'summit',
'helseggen', 'storm', 'description', 'impression', 'spectacle', 'moskoe',
'depth', 'water', 'thirty', 'vurrgh', 'depth', 'passage', 'vessel', 'risk',
'stream', 'country', 'rapidity', 'ebb', 'sea', 'scarce', 'ship', 'attraction',
'beat', 'water', 'relaxes', 'tranquility', 'flood', 'calm', 'quarter', 'hour',
'violence', 'stream', 'fury', 'reach', 'stream', 'violence', 'moskoe', 'borne',
'pine', 'rise', 'broken', 'degree', 'craggy', 'stream', 'reflux', 'sea',
'water', 'year', 'morning', 'sexagesima', 'sunday', 'impetuosity', 'water',
'vicinity', 'vortex', 'reference', 'moskoe', 'depth', 'centre', 'moskoe', 'str',
'proof', 'fact', 'glance', 'abyss', 'whirl', 'crag', 'helseggen', 'pinnacle',
'phlegethon', 'simplicity', 'jonas', 'ramus', 'belief', 'fact', 'thing', 'ship',
'existence', 'influence', 'resist', 'hurricane', 'remember', 'perusal',
'aspect', 'idea', 'collision', 'reflux', 'ridge', 'water', 'flood', 'fall',
'result', 'whirlpool', 'vortex', 'suction', 'lesser', 'dia', 'britannica',
'imagine', 'centre', 'channel', 'maelstr', 'globe', 'gulf', 'bothnia',
'instance', 'opinion', 'imagination', 'guide', 'view', 'notion', 'inability',
'paper', 'absurd', 'thunder', 'abyss', 'whirl', 'man', 'round', 'crag', 'lee',
'roar', 'water', 'story', 'convince', 'moskoe', 'str', 'smack', 'seventy',
'habit', 'fishing', 'fishing', 'business', 'southward', 'fish', 'risk',
'choice', 'variety', 'abundance', 'day', 'craft', 'scrape', 'week', 'fact',
'matter', 'speculation', 'risk', 'life', 'labor', 'courage', 'capital', 'smack',
'cove', 'coast', 'practice', 'fine', 'weather', 'advantage', 'push', 'channel',
'moskoe', 'pool', 'drop', 'anchorage', 'sandflesen', 'time', 'water',
'expedition', 'wind', 'return', 'seldom', 'calculation', 'point', 'night',
'anchor', 'account', 'calm', 'thing', 'remain', 'week', 'death', 'gale',
'occasion', 'sea', 'spite', 'round', 'anchor', 'cross', 'day', 'luck', 'spot',
'weather', 'shift', 'gauntlet', 'moskoe', 'accident', 'heart', 'mouth', 'slack',
'wind', 'smack', 'brother', 'son', 'assistance', 'risk', 'heart', 'danger',
'danger', 'truth', 'day', 'day', 'hurricane', 'morning', 'afternoon', 'breeze',
'south', 'sun', 'shone', 'seaman', 'foreseen', 'clock', 'fish', 'day', 'str',
'slack', 'water', 'wind', 'quarter', 'time', 'rate', 'danger', 'reason',
'aback', 'breeze', 'helseggen', 'boat', 'wind', 'headway', 'return',
'anchorage', 'horizon', 'copper', 'cloud', 'velocity', 'breeze', 'direction',
'state', 'time', 'minute', 'storm', 'sky', 'spray', 'hurricane', 'seaman',
'thing', 'board', 'brother', 'safety', 'boat', 'thing', 'water', 'flush',
'hatch', 'bow', 'hatch', 'custom', 'str', 'precaution', 'circumstance', 'lay',
'brother', 'destruction', 'opportunity', 'threw', 'deck', 'bow', 'foot', 'fore',
'mast', 'instinct', 'thing', 'time', 'breath', 'clung', 'bolt', 'stand', 'dog',
'water', 'measure', 'arm', 'elder', 'brother', 'heart', 'joy', 'moment', 'joy',
'ear', 'word', 'moskoe', 'moment', 'shook', 'head', 'foot', 'ague', 'word',
'understand', 'wind', 'crossing', 'channel', 'calmest', 'weather', 'wait',
'watch', 'pool', 'hurricane', 'moment', 'dream', 'hope', 'time', 'fury',
'spent', 'feel', 'change', 'direction', 'pitch', 'overhead', 'burst', 'rift',
'sky', 'bright', 'moon', 'lustre', 'thing', 'distinctness', 'scene', 'brother',
'manner', 'din', 'word', 'voice', 'head', 'death', 'thought', 'watch', 'fob',
'face', 'moonlight', 'burst', 'clock', 'time', 'slack', 'whirl', 'fury', 'boat',
'laden', 'slip', 'beneath', 'landsman', 'sea', 'phrase', 'ridden', 'sea',
'bore', 'rise', 'sweep', 'slide', 'plunge', 'feel', 'dizzy', 'mountain',
'dream', 'glance', 'glance', 'position', 'instant', 'moskoe', 'quarter', 'mile',
'day', 'whirl', 'race', 'place', 'horror', 'spasm', 'boat', 'half', 'direction',
'thunderbolt', 'moment', 'water', 'kind', 'shrill', 'waste', 'steam', 'steam',
'belt', 'surf', 'moment', 'plunge', 'abyss', 'velocity', 'boat', 'sink',
'water', 'air', 'surface', 'surge', 'whirl', 'larboard', 'wall', 'horizon',
'mind', 'hope', 'deal', 'terror', 'despair', 'truth', 'thing', 'die', 'manner',
'paltry', 'consideration', 'life', 'view', 'manifestation', 'god', 'power',
'idea', 'mind', 'curiosity', 'whirl', 'sacrifice', 'grief', 'man', 'mind',
'extremity', 'boat', 'pool', 'light', 'circumstance', 'possession', 'cessation',
'reach', 'situation', 'surf', 'bed', 'ocean', 'ridge', 'sea', 'form', 'idea',
'confusion', 'mind', 'spray', 'blind', 'deafen', 'strangle', 'power', 'action',
'reflection', 'measure', 'death', 'prison', 'doom', 'circuit', 'belt', 'round',
'hour', 'surge', 'nearer', 'nearer', 'edge', 'time', 'bolt', 'brother', 'water',
'cask', 'thing', 'deck', 'brink', 'pit', 'agony', 'terror', 'force', 'secure',
'grasp', 'grief', 'attempt', 'sheer', 'care', 'contest', 'point', 'difference',
'cask', 'difficulty', 'smack', 'sweeps', 'position', 'starboard', 'prayer',
'god', 'descent', 'hold', 'barrel', 'destruction', 'death', 'water', 'moment',
'moment', 'sense', 'motion', 'vessel', 'exception', 'courage', 'scene',
'forget', 'horror', 'admiration', 'boat', 'surface', 'funnel', 'circumference',
'depth', 'bewildering', 'rapidity', 'spun', 'radiance', 'shot', 'rift', 'glory',
'observe', 'burst', 'terrific', 'grandeur', 'beheld', 'gaze', 'direction',
'view', 'manner', 'smack', 'hung', 'pool', 'keel', 'deck', 'plane', 'water',
'angle', 'beam', 'difficulty', 'situation', 'level', 'speed', 'search',
'profound', 'gulf', 'mist', 'magnificent', 'rainbow', 'bridge', 'time',
'eternity', 'mist', 'spray', 'doubt', 'yell', 'attempt', 'slide', 'abyss',
'distance', 'slope', 'proportionate', 'round', 'round', 'movement', 'circuit',
'whirl', 'progress', 'revolution', 'waste', 'ebony', 'borne', 'boat', 'object',
'embrace', 'whirl', 'house', 'furniture', 'curiosity', 'place', 'drew',
'nearer', 'nearer', 'doom', 'company', 'amusement', 'time', 'thing', 'plunge',
'wreck', 'dutch', 'merchant', 'ship', 'overtook', 'length', 'making', 'fact',
'fact', 'miscalculation', 'train', 'reflection', 'heart', 'dawn', 'hope',
'memory', 'observation', 'variety', 'matter', 'moskoe', 'str', 'number',
'chafed', 'account', 'difference', 'supposing', 'whirl', 'period', 'reason',
'reach', 'turn', 'flood', 'ebb', 'case', 'instance', 'fate', 'rule', 'shape',
'superiority', 'speed', 'descent', 'size', 'shape', 'cylinder', 'school',
'master', 'district', 'explanation', 'fact', 'consequence', 'vortex',
'resistance', 'suction', 'drawn', 'difficulty', 'body', 'form', 'circumstance',
'turn', 'account', 'revolution', 'barrel', 'yard', 'mast', 'level', 'station',
'lash', 'water', 'cask', 'counter', 'throw', 'water', 'attention', 'power',
'length', 'design', 'case', 'shook', 'head', 'station', 'reach', 'emergency',
'delay', 'bitter', 'struggle', 'fate', 'sea', 'moment', 'hesitation', 'result',
'escape', 'possession', 'mode', 'escape', 'anticipate', 'story', 'conclusion',
'hour', 'smack', 'distance', 'beneath', 'succession', 'chaos', 'barrel', 'sunk',
'half', 'distance', 'gulf', 'spot', 'change', 'place', 'character', 'whirlpool',
'slope', 'rainbow', 'sky', 'view', 'spot', 'pool', 'moskoe', 'hour', 'slack',
'sea', 'str', 'fatigue', 'danger', 'memory', 'horror', 'board', 'traveller',
'spirit', 'land', 'hair', 'day', 'expression', 'countenance', 'story', 'faith'],
'DT': [], 'PRP': [], ':': [], 'WP$': [], 'NNPS': [], 'PRP$': [], 'WDT': [], '(':
[], ')': [], '.': [], ',': [], '``': [], '$': [], 'RB': ['depth', 'long', 'ago',
'deadly', 'scarcely', 'carelessly', 'beneath', 'deeply', 'length', 'upward',
'dizzily', 'deplorably', 'horridly', 'forcibly', 'ghastly', 'forever',
'properly', 'nearer', 'hideously', 'constantly', 'midway', 'northward',
'gradually', 'rapidly', 'eastward', 'vurrgh', 'sway', 'suddenly', 'suddenly',
'suddenly', 'dizzily', 'heaven', 'excess', 'exceedingly', 'weather', 'moskoe',
'noise', 'heard', 'inevitably', 'gradually', 'storm', 'norway', 'likewise',
'frequently', 'terribly', 'shore', 'plainly', 'constantly', 'early', 'ground',
'regard', 'close', 'immeasurably', 'deadly', 'bodily', 'generally', 'decidedly',
'universally', 'altogether', 'schooner', 'shortly', 'violently', 'stout',
'afterward', 'brightly', 'suddenly', 'folly', 'norway', 'cleverly', 'mainmast',
'completely', 'presently', 'overboard', 'horror', 'bound', 'long', 'carefully',
'slack', 'ear', 'presently', 'properly', 'cleverly', 'presently', 'ahead',
'involuntarily', 'afterward', 'suddenly', 'subside', 'completely',
'indistinctly', 'positively', 'considerably', 'gradually', 'securely',
'overboard', 'steadily', 'scarcely', 'abyss', 'felt', 'instinctively', 'midway',
'perfectly', 'ghastly', 'accurately', 'instinctively', 'scarcely', 'distinctly',
'nature', 'heavily', 'arose', 'partly', 'partly', 'mind', 'appearance',
'distinctly', 'completely', 'slowly', 'early', 'rapidly', 'slowly', 'sphere',
'equally', 'longer', 'securely', 'despairingly', 'counter', 'precisely',
'brother', 'headlong', 'forever', 'farther', 'overboard', 'momently',
'gradually', 'slowly', 'moon', 'radiantly', 'borne', 'violently', 'scarcely'],
'RBR': ['shadow', 'farther', 'matter', 'feather', 'higher', 'longer', 'listen',
'explore', 'farther', 'limbs', 'cylinder', 'farther'], 'RBS': [], 'VBD':
['reached', 'exhausted', 'guided', 'happened', 'happened', 'survived',
'frightened', 'hung', 'arose', 'tempted', 'excited', 'fell', 'dared',
'struggled', 'mentioned', 'continued', 'distinguished', 'looked', 'wore',
'left', 'lay', 'illustrated', 'enveloped', 'encompassed', 'shore', 'plunged',
'resumed', 'keildhelm', 'thought', 'spoke', 'perceived', 'held', 'seamed',
'grew', 'disappeared', 'subsided', 'represented', 'slipped', 'inclined',
'trembled', 'rocked', 'termed', 'surveyed', 'quoted', 'equalled', 'absorbed',
'heightened', 'carried', 'overpowered', 'caught', 'roared', 'absorbed', 'grew',
'whirled', 'regulated', 'raged', 'fell', 'ascertained', 'appeared', 'wore',
'named', 'assented', 'surprised', 'entertained', 'desired', 'owned', 'weighed',
'felt', 'forced', 'blew', 'threw', 'dragged', 'brought', 'encountered',
'happened', 'thought', 'occurred', 'july', 'crossed', 'loaded', 'remarked',
'started', 'knew', 'spanked', 'happened', 'began', 'covered', 'colored', 'rose',
'fell', 'becalmed', 'blew', 'sawed', 'lashed', 'sat', 'foundered', 'prompted',
'flurried', 'deluged', 'raised', 'felt', 'leaped', 'turned', 'screamed', 'knew',
'knew', 'drove', 'thought', 'cursed', 'knew', 'scudded', 'knew', 'increased',
'screamed', 'held', 'flashed', 'glanced', 'built', 'called', 'happened', 'rose',
'believed', 'moskoe', 'closed', 'clenched', 'felt', 'enveloped', 'drowned',
'wore', 'skim', 'arose', 'left', 'composed', 'unmanned', 'began', 'blushed',
'crossed', 'felt', 'thought', 'rendered', 'tended', 'belt', 'towered',
'occasioned', 'allowed', 'lashed', 'swept', 'endeavored', 'felt', 'knew',
'fright', 'knew', 'held', 'flew', 'rushed', 'muttered', 'thought', 'tightened',
'closed', 'dared', 'wondered', 'elapsed', 'belt', 'lay', 'looked', 'gazed',
'appeared', 'streamed', 'recovered', 'fell', 'inclined', 'lay', 'sloped',
'suppose', 'enveloped', 'occasioned', 'belt', 'carried', 'swept', 'perceived',
'appeared', 'began', 'sought', 'disappointed', 'beat', 'strewed', 'shattered',
'roughened', 'recollected', 'roughened', 'absorbed', 'descended', 'conceived',
'whirled', 'absorbed', 'learned', 'observed', 'happened', 'offered', 'passed',
'opened', 'hesitated', 'held', 'attracted', 'pointed', 'comprehended',
'refused', 'admitted', 'resigned', 'hoped', 'effected', 'descended', 'loved',
'attached', 'leaped', 'grew', 'disappeared', 'str', 'heaved', 'hurried',
'removed', 'drew', 'knew', 'changed'], 'IN': ['broken', 'teeth', 'otterholm',
'ver', 'abyss', 'amid', 'drove', 'overcast', 'thrown', 'wind', 'wind', 'round',
'round', 'wild', 'amid', 'tide', 'thereabout'], 'FW': ['elbow', 'kircher'],
'RP': [], 'JJR': ['greater', 'weightier', 'higher', 'smaller', 'yonder',
'greater', 'smaller', 'higher', 'deeper', 'lower', 'greater', 'counter',
'restore', 'lower', 'deeper', 'smaller', 'greater', 'larger', 'cylinder',
'greater', 'brother'], 'JJS': ['loftiest', 'youngest', 'divest', 'crest',
'faintest', 'loudest', 'highest', 'honest', 'largest', 'finest', 'eldest',
'west', 'oldest', 'worst', 'slightest', 'oldest', 'youngest', 'lightest',
'tempest', 'greatest', 'keenest', 'west'], 'PDT': [], 'MD': [], 'VB': ['raise',
'timid', 'morrow', 'watch', 'deck', 'elder', 'shake', 'slack', 'keel', 'hold',
'slow', 'channel'], 'WRB': [], 'NNP': [], 'EX': [], 'NNS': ['ways', 'ways',
'models', 'minutes', 'sons', 'years', 'hours', 'limbs', 'nerves', 'feet',
'crags', 'yards', 'foundations', 'winds', 'fancies', 'waters', 'ramparts',
'lines', 'miles', 'miles', 'barren', 'intervals', 'rocks', 'rocks',
'norwegians', 'flimen', 'names', 'places', 'minutes', 'buffaloes', 'minutes',
'waters', 'channels', 'vortices', 'descents', 'minutes', 'whirlpools',
'streaks', 'streaks', 'vortices', 'degrees', 'winds', 'lifts', 'norwegians',
'accounts', 'jonas', 'confounds', 'details', 'fathoms', 'rocks', 'cataracts',
'vortices', 'pits', 'pieces', 'fragments', 'intervals', 'boats', 'yachts',
'ships', 'howlings', 'bellowings', 'struggles', 'stocks', 'trees', 'bristles',
'rocks', 'hours', 'stones', 'houses', 'fathoms', 'portions', 'records',
'anecdotes', 'bears', 'attempts', 'vortices', 'islands', 'waves', 'shelves',
'confines', 'experiments', 'norwegians', 'brothers', 'tons', 'islands',
'eddies', 'opportunities', 'coastmen', 'islands', 'grounds', 'hours', 'places',
'spots', 'rocks', 'miles', 'minutes', 'eddies', 'years', 'grounds', 'channel',
'whirlpools', 'currents', 'flimen', 'difficulties', 'grounds', 'times', 'years',
'times', 'days', 'years', 'people', 'heavens', 'gentle', 'brothers', 'islands',
'eddies', 'things', 'sails', 'masts', 'moments', 'feet', 'hands', 'moments',
'knees', 'hands', 'seas', 'senses', 'feelings', 'str', 'times', 'events',
'mountains', 'attempts', 'fingers', 'tears', 'waves', 'swells', 'eyes', 'lids',
'minutes', 'waves', 'pipes', 'vessels', 'jaws', 'felt', 'nerves', 'companions',
'mysteries', 'fancies', 'revolutions', 'annoyances', 'felons', 'indulgences',
'hands', 'swelters', 'eyes', 'seconds', 'struggles', 'sensations', 'sides',
'rays', 'clouds', 'recesses', 'degrees', 'rays', 'mussulmen', 'walls',
'heavens', 'swings', 'jerks', 'yards', 'fragments', 'vessels', 'masses',
'trunks', 'trees', 'articles', 'pieces', 'boxes', 'barrels', 'terrors',
'things', 'velocities', 'descents', 'disappears', 'guesses', 'articles',
'splinters', 'fragments', 'observations', 'bodies', 'masses', 'conversations',
'forms', 'fragments', 'observations', 'things', 'eyes', 'wonders', 'signs',
'barrels', 'bolt', 'lashings', 'gyrations', 'sides', 'funnel', 'gyrations',
'degrees', 'winds', 'shores', 'waves', 'effects', 'minutes', 'grounds',
'fishermen', 'mates', 'companions', 'fishermen'], 'SYM': [], 'CC': [], 'CD': [],
'POS': []}
```

0.4.4 Question 4: Build a dataframe to store the text and title of the story, along with columns representing the two-letter prefixes of each tag

[100]:
```
from pyspark.sql.types import StructType, StructField, StringType, ArrayType
from functools import reduce
from copy import deepcopy
import os

def append_if_starts_with(prefix_dict, dict_tuple):
  '''
  Helper function for reducer that converts the tag dictionary to one where
  only certain two-letter prefixes are considered, and all categories with
  that prefix are combined
  '''
  prefix_list=['NN', 'VB', 'JJ', 'RB']
  pos, word_list = dict_tuple
  new_dict = deepcopy(prefix_dict)
  if len(pos) >= 2 and pos[:2] in prefix_list:
    prefix = pos[:2]
    if prefix in prefix_dict:
        old_val = prefix_dict[prefix]
    else:
        old_val = []
    return new_dict | {prefix: old_val + word_list}
  else:
    return new_dict

def convert_to_prefix_dict(tag_dict):
    '''
    Converts a tag dictionary to a new dicitonary that combines values by
    two-letter prefix of the key
    '''

    return reduce(append_if_starts_with, list(tag_dict.items()), {})


def add_book(df, title):
  '''
  Adds the information of a book to a row of a distributed dataframe
  '''
  paragraph = clean_book(title)
  tag_dict = get_tag_dict(paragraph, taginfo)
  prefix_dict = convert_to_prefix_dict(tag_dict)
  schema = StructType([StructField(key, ArrayType(StringType()), True) for key␣
 ↪in prefix_dict])
  result_df = spark.createDataFrame([prefix_dict], schema=schema) \
    .withColumn('Title', lit(title.replace("_"," ").title())) \
    .withColumn('text', lit(paragraph))
  if df is None:
    return result_df
  else:
    return df.union(result_df)

# Add all books in folder to dataframe
df = None
for book in os.listdir('big-data-repo/text-proc/poe-stories/'):
  df = add_book(df, book)

# This displays better in the ipynb file:
df.show()
```

```
+--------------------+--------------------+--------------------+----------------
----+--------------------+--------------------+
|                  VB|                  JJ|                  NN|
RB|               Title|                text|
+--------------------+--------------------+--------------------+----------------
----+--------------------+--------------------+
|[published, hurri…|[elaborate, summa…|[minute, paper, j…|[simply, ago,
spe…|Von Kempelen And …|minute elaborate …|
|[honored, connect…|[dicebant, amicae…|[mihi, visitarem,…|[intimately,
last…|            Berenice|dicebant mihi sod…|
|[purloined, maint…|[odiosius, seneca…|[letter, nil, sap…|[observer,
intent…|The Purloined Letter|purloined letter …|
|[fallen, risen, c…|[fail, vale, exeq…|[stay, meet, deat…|[youth, art,
sure…|      The Assignation|stay fail meet th…|
|[enjoyed, estimat…|[nullus, enim, mo…|[locus, sine, gen…|[mockery, lui,
fu…|The Island Of The…|nullus enim locus…|
|[undulated, hit, …|[arnheim, pedestr…|[pendant, domain,…|[remarkably,
conf…|     Landors Cottage|pendant domain ar…|
|[selected, endure…|[horrible, legiti…|[fiction, romanti…|[regard,
vividly,…|The Premature Burial|themes absorbing …|
|[set, fathomed, b…|[prima, human, ra…|[consideration, s…|[equally,
solely,…|The Imp Of The Pe…|consideration fac…|
|[concerned, rende…|[pretend, extraor…|[matter, case, va…|[naturally,
succi…|The Facts In The …|pretend matter ex…|
|[paused, pondered…|[suspendu, dull, …|[son, luth, sit, …|[oppressively,
si…|The Fall Of The H…|son luth suspendu…|
|[quoted, pardoned…|[oriental, tellme…|[truth, stranger,…|[scarcely,
discov…|The Thousand-And-…|truth stranger fi…|
|[refined, suppose…|[rationale, mere,…|[doubt, mesmerism…|[universally,
mer…| Mesmeric Revelation|doubt envelop rat…|
|[exalted, loved, …|[specific, anima,…|[conservatione, f…|[lully, wisdom,
a…|            Eleonora|conservatione for…|
|[resolved, contin…|[borne, ventured,…|[insult, soul, su…|[definitively,
eq…|The Cask Of Amont…|injuries fortunat…|
|[agitated, fallen…|[silent, listen, …|[mountain, hand, …|[forever,
forever…|     Silence-A Fable|mountain pinnacle…|
|[endured, beheld,…|[nature, commensu…|[providence, vast…|[depth, long,
ago…|A Descent Into Th…|ways god nature p…|
|[resolved, red, e…|[red, devastated,…|[death, country, …|[long, august,
fo…|The Masque Of The…|red death long de…|
|[thought, feel, s…|[longos, hic, inn…|[impia, tortorum,…|[long,
presently,…|The Pit And The P…|impia tortorum lo…|
|[scorn, bruited, …|[grim, spectre, f…|[conscience, path…|[art, forever,
et…|      William Wilson|conscience grim s…|
|[tortured, derive…|[wild, narrative,…|[belief, mad, cas…|[homely, surely,
…|       The Black Cat|wild homely narra…|
+--------------------+--------------------+--------------------+----------------
----+--------------------+--------------------+
only showing top 20 rows
```

0.4.5 Question 5: Determine if the number of different parts of speech

[101]:
```python
from pyspark.sql.functions import size

p3q5_df = df \
  .withColumn('Total_Words', size(f.split('text', " "))) \
  .withColumn('Verbs_per_1000', f.round(1000*size('VB')/col('Total_Words'),0).
 ↪cast('integer')) \
  .withColumn('Nouns_per_1000', f.round(1000*size('NN')/col('Total_Words'),0).
 ↪cast('integer')) \
  .withColumn('Adjectives_per_1000', f.round(1000*size('JJ')/
 ↪col('Total_Words'),0).cast('integer')) \
  .withColumn('Adverbs_per_1000', f.round(1000*size('RB')/col('Total_Words'),0).
↪cast('integer')) \
  .drop('VB', 'NN', 'JJ', 'RB', 'text', 'Total_Words')


p3q5_df.show(n=df.count(), truncate=False)
```

```
+---------------------------------------------+--------------+--------------+----
---------------+----------------+
|Title
|Verbs_per_1000|Nouns_per_1000|Adjectives_per_1000|Adverbs_per_1000|
+---------------------------------------------+--------------+--------------+----
---------------+----------------+
|Von Kempelen And His Discovery              |219           |472           |236
|64              |
|Berenice                                    |219           |471           |242
|62              |
|The Purloined Letter                        |221           |507           |221
|44              |
|The Assignation                             |226           |484           |235
|49              |
|The Island Of The Fay                       |218           |459           |251
|61              |
|Landors Cottage                             |194           |457           |275
|67              |
|The Premature Burial                        |229           |463           |241
|63              |
|The Imp Of The Perverse                     |206           |497           |235
|55              |
|The Facts In The Case Of M. Valdemar        |219           |466           |243
|69              |
|The Fall Of The House Of Usher              |219           |468           |243
|63              |
|The Thousand-And-Second Tale Of Scheherazade|222           |495           |224
|51              |
|Mesmeric Revelation                         |177           |516           |246
|53              |
|Eleonora                                    |236           |475           |224
|59              |
|The Cask Of Amontillado                     |222           |473           |240
|59              |
|Silence-A Fable                             |230           |479           |215
|67              |
|A Descent Into The Maelstrom                |215           |492           |221
|64              |
|The Masque Of The Red Death                 |221           |501           |221
|53              |
|The Pit And The Pendulum                    |245           |478           |212
```

```
|60               |
|William Wilson                                   |216            |473            |245
|58               |
|The Black Cat                                    |220            |485            |234
|56               |
|The Domain Of Arnheim                            |198            |495            |249
|51               |
+--------------------------------------------+--------------+--------------+----
---------------+-----------------+
```

[102]:
```
from scipy.stats import variation

verbs_cv = variation([row['Verbs_per_1000'] for row in p3q5_df.collect()])
nouns_cv = variation([row['Nouns_per_1000'] for row in p3q5_df.collect()])
adjs_cv = variation([row['Adjectives_per_1000'] for row in p3q5_df.collect()])
advs_cv = variation([row['Adverbs_per_1000'] for row in p3q5_df.collect()])

print("Verbs Coefficient of Variation: " + str(verbs_cv))
print("Nouns Coefficient of Variation: " + str(nouns_cv))
print("Adjectives Coefficient of Variation: " + str(adjs_cv))
print("Adverbs Coefficient of Variation: " + str(advs_cv))
```

```
Verbs Coefficient of Variation: 0.06536039141700235
Nouns Coefficient of Variation: 0.03279240652582868
Adjectives Coefficient of Variation: 0.060363064825201665
Adverbs Coefficient of Variation: 0.11002228396917048
```

The highest coefficient of variation among all 4 parts of speech is only .11, which means its standard deviation is only 11% as large as its mean. I would consider this to be very small, which is consistent with the conjecture that we would expect the frequency of occurence of each part of speech to be approximately the same throughout each book.