IST 664
Natural Language Processing

Assignment:
Homework 2

Rashmitha Varma Pandati
SUID: 622666081

Table of Contents

REGULAR EXPRESSIONS TO ANALYZE NATIONAL SCIENCE FOUNDATION ABSTRACTS DATA

1. Introduction

The National Science Foundation Research Awards Abstracts corpus makes an audit of collection of different aspects of research and grants awarded for research being carried out across all domains and fields of study.

2. Pre-processing

A. Corpus Characteristics

The publicly available open source corpus collection of NSF research awards abstracts spanning 1990 – 2003 is analyzed as follows:

- **History**

  The type of data consisting in this corpus is of text and tabular type. This corpus is the work from UCI [University of California, Irvine] and is originally owned by Michael J. Pazzani, who provided the abstracts, and Amnon Meyers, who provided the bag of words. These were donated on November 18, 2003.

- **Data Description**

  The NSF Research Awards Abstracts have three main documents and they are:

  - **134,161 abstracts describing NSF awards for basic research**

    All the abstracts, one per file, were furnished by the NSF (National Science Foundation).

  - **Bag-of-word data files extracted from the abstracts**

    The bag-of-word data was produced by automatically processing the abstracts with a text analyzer called NSFAbst, built using VisualText. While most fields of the output are very accurate, the authors were not extracted from the Investigator field with 100% accuracy, due to wide variability.

  - **List of words used for indexing the bag-of-word data**

    The word list came from a separate process and may not include all the words of interest in the abstracts.

- **Number of Documents**

  There are total of 4016 awards abstracts in the NSF Awards Abstracts dataset and its folder structure is as follows:

  ```
  | RootDirectory
      | \ NSF_abstracts
          | | \ Text File(.txt)
  ```

- **Naming Convention**

  The files have been named in the format '*.txt' where * represents the unique ID generated as the file was processed. The files consist of information like Title, File, Award Grant, Award department, Award start and end date, Sponsor etc. as fields followed by its values. For example:

- "Title" : "Genetic Diversity of Endangered Populations of Mysticete Whales: Mitochondrial DNA and Historical Demography"
- "File" : "a9000006"
- "Award Number": "9000006"
- "Sponsor" : "U of Hawaii Manoa".

B. Reading and Extraction of Abstracts

- To overcome the opened file descriptor for reading and then extracting necessary items from large entries of directories, we copy the contents of each abstract to a list. We use list comprehension with Pathlib library to process the directory entries fast and efficient in object -oriented manner.
- Since, the fields are identical across all abstracts, we use regular expression to extract from them.
  - File/AbstractID : *" **File.*:(.*)** "* - To extract File ID. The regular expression extracts any character followed by "File :" field.
  - NSF Organization: *" **NSF Org.*:(.*)** "* - To extract NSF Org type. The regular expression extracts any character followed by "NSF Org :" field.
  - Award Amount : "***Total Amt.*(\$[0-9]*)***" - To extract award grant. The regular expression extracts any digits followed by "Total Amt. :" field.
  - Abstract : "***\bAbstract\b.*([\s\S]+)***" - To extract the abstract field. The regular expression matches any character (digits, alphanumeric, special, newline, tab, return, multiline) followed by "Abstract :" field.
- Once we extract the data using regular expression , we print them in tabular format and write to the text file – "TabularOutput.txt".

3. Distribution of Sentence Lengths

- Once we have extracted the fields and values from each abstract, we perform tokenization using nltk tokenizer.
- We choose ***Punkt*** library for sentence tokenization. ***Punkt.PunktTokenizer*** provides a way of handling special characters like period within respective words (such as prefix) and actual end of sentence.
  - For eg: "Dr. Smith" is not end of sentence but a prefix.
- Each token identifies a sentence among each abstract. We print the details of sentence like sentence number, belonging file, corresponding sentence. Each of these fields are separated by **"|"** symbol.
- We also print number of sentences for first 3 abstracts in tabular format on console and each abstracts are written to file "SentenceOutput.txt".

4. Output Observations

A) TabularOutput.txt

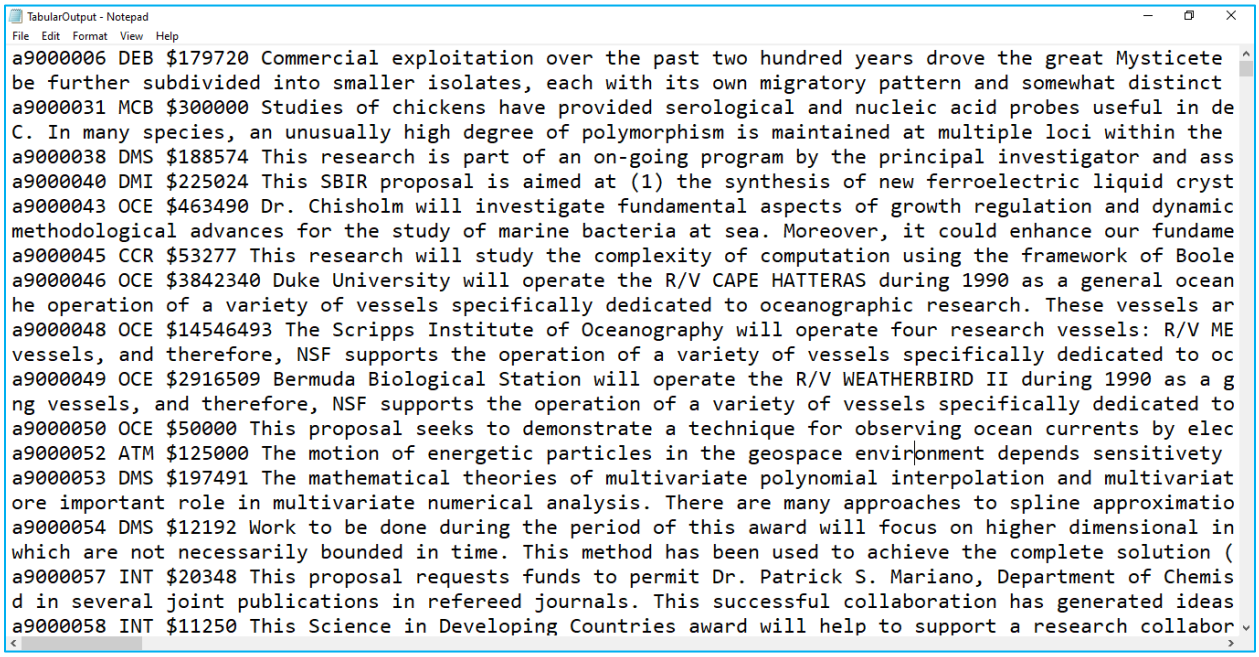
The required details namely, File Identity, NSF Org type, Total Amount and Abstract Content, of all the 4016 documents, are displayed in tabular format and saved in the file.

B) SentenceOutput.txt

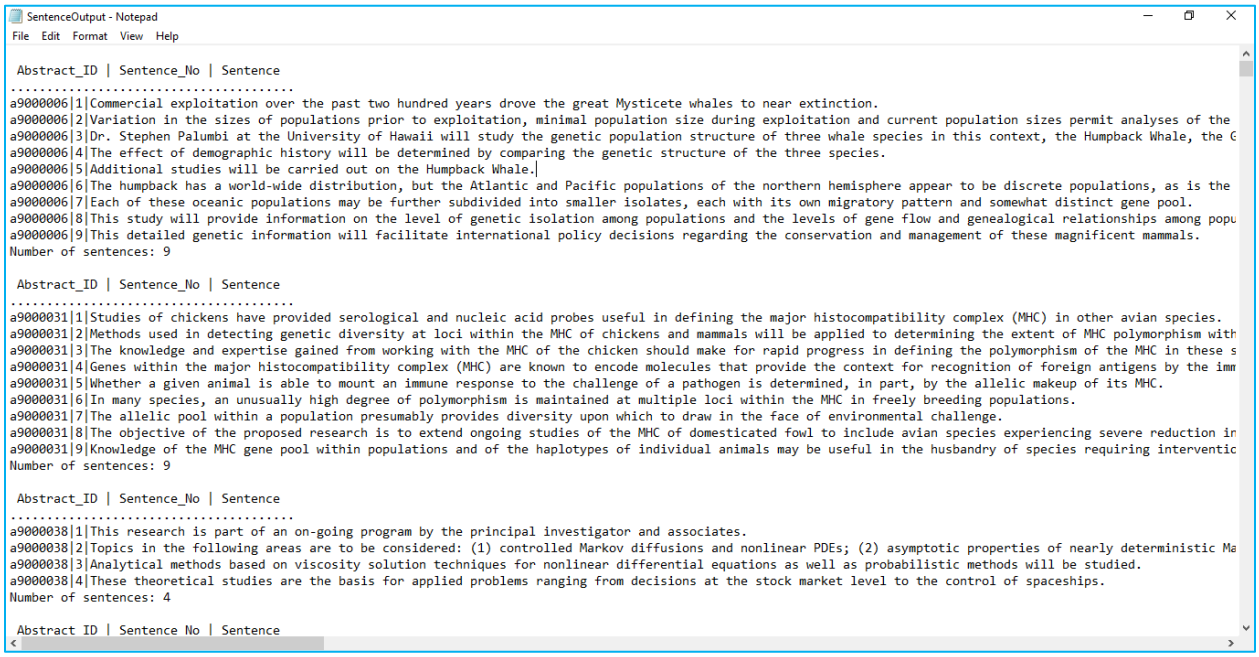
The effect of sentence tokenization is shown by the display of each sentence individually along with the total number of sentences in that abstract.

5. Appendix One: Outputs

A) Tablular Output



B) Sentence Output



6. Appendix Two: Python Code

```
#importing packages
import nltk
import sys
import os
import zipfile
from zipfile import ZipFile
import glob
import nltk
import re
import pathlib


#Initializing the folder structure
path = pathlib.Path("C:/Users/girl1/OneDrive/Documents/664/Homework 2/NSF_abstracts").resolve().parent
NSFPath = path/"NSF_abstracts"


#Reading the contents of the text files present in the folder
AbstractRead = [file.read_text(encoding="ISO-8859-1") for file in NSFPath.rglob('*.txt')]


#Declaring an Empty List
AbstractStore = []
#Looping the extarction functions together for all the abstract files
for text in AbstractRead:
    if text:
        #Extracting the Filename from the abstract
        Abstract_Identity = re.findall(r"File.*:(.*)", text)[0].strip()
        #Extracting the Organization Type from the abstract
        NSFOrgType = re.findall(r"NSF Org.*:(.*)", text)[0].strip()
        #Extracting the Amount from the abstarct
        Amount=re.findall(r"Total Amt.*(\$[0-9]*)", text)[0].strip()
        #Extracting the content from the abstract
        AbstractContent = " ".join(re.findall(r"\bAbstract\b.*([\s\S]+)", text)[0].split())
        #Appending all the obtained outputs into the empty list
        AbstractStore.append({"File":Abstract_Identity,"OrgType":NSFOrgType,"Amount":Amount,"Abstract":AbstractContent})


#Printing the new list
print(AbstractStore)


#Printing the output into a text file
with open('TabularOutput.txt',"w+", encoding = 'UTF-8') as file:
            for item in AbstractStore:
                Data = item["File"]+" "+item["OrgType"]+" "+item["Amount"]+" "+item["Abstract"]
                file.write(Data+"\n")
                print(Data)


#Declaring the punkt sentence tokenizer
SentenceDetector = nltk.data.load('tokenizers/punkt/english.pickle')
#Initializing count
```

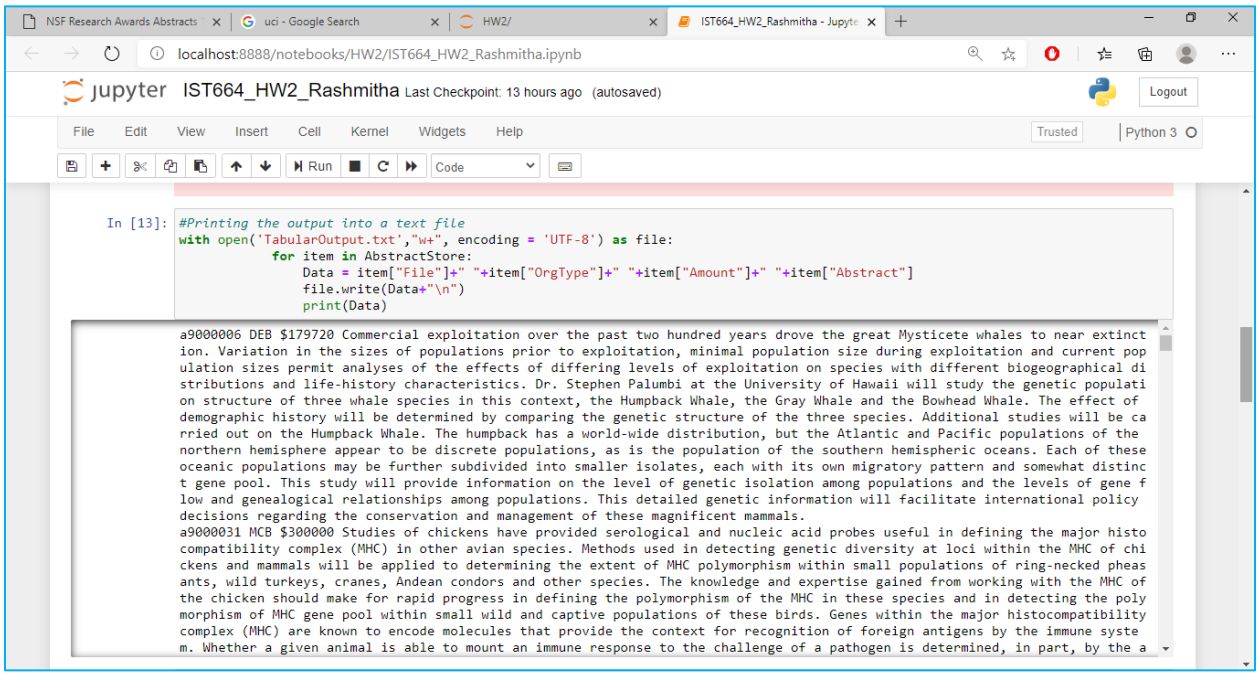
```
count = 0
print("\n Sentence Tokenizer\n")
#Printing the output into a text file
with open("SentenceOutput.txt","w+", encoding = 'UTF-8') as file:
    for item in AbstractStore:
        #Distribution of Sentence Lengths for the first 3 abstracts
        if count < 3:
            print(" Abstract_ID | Sentence_No | Sentence ")
            print(".....................................")

            file.write("\n Abstract_ID | Sentence_No | Sentence \n.....
.....................\n")
            SentenceList = SentenceDetector.tokenize(item["Abstract"].strip())
            for idx,sentence in enumerate(SentenceList):
                print(item["File"]+"|"+str(idx+1)+"|"+sentence)
                file.write(item["File"]+"|"+str(idx+1)+"|"+sentence+"\n")
            print("Number of sentences: ",len(SentenceList))
            file.write("Number of sentences: "+str(len(SentenceList))+"\n")
            print()
            count = count + 1
        #Distribution of Sentence Lengths for remaining abstracts
        else:
            file.write("\n Abstract_ID | Sentence_No | Sentence \n.....
.....................\n")
            SentenceList = SentenceDetector.tokenize(item["Abstract"].strip())
            for idx, sentence in enumerate(SentenceList):
                file.write(item["File"] + "|" + str(idx + 1) + "|" + sentence + "\n")
            file.write("Number of sentences: " + str(len(SentenceList)) + "\n")

```

7. Appendix Three: IDE Screenshots

A) Tabular Output



B) Sentence Output [For 3 abstracts]

```
In [15]: #Declaring the punkt sentence tokenizer
SentenceDetector = nltk.data.load('tokenizers/punkt/english.pickle')
#Initializing count
count = 0
print("\n Sentence Tokenizer\n")
#Printing the output into a text file
with open("SentenceOutput.txt","w+", encoding = 'UTF-8') as file:
    for item in AbstractStore:
        #Distribution of Sentence Lengths for the first 3 abstracts
        if count < 3:
            print(" Abstract_ID | Sentence_No | Sentence ")
            print(".....................................")

            file.write("\n Abstract_ID | Sentence_No | Sentence \n.....................................\n")
            SentenceList = SentenceDetector.tokenize(item["Abstract"].strip())
            for idx,sentence in enumerate(SentenceList):
                print(item["File"]+"|"+str(idx+1)+"|"+sentence)
                file.write(item["File"]+"|"+str(idx+1)+"|"+sentence+"\n")
            print("Number of sentences: ",len(SentenceList))
            file.write("Number of sentences: "+str(len(SentenceList))+"\n")
            print()
            count = count + 1
        #Distribution of Sentence Lengths for remaining abstracts
        else:
            file.write("\n Abstract_ID | Sentence_No | Sentence \n.....................................\n")
            SentenceList = SentenceDetector.tokenize(item["Abstract"].strip())
            for idx, sentence in enumerate(SentenceList):
                file.write(item["File"] + "|" + str(idx + 1) + "|" + sentence + "\n")
            file.write("Number of sentences: " + str(len(SentenceList)) + "\n")
```

```
 Sentence Tokenizer

 Abstract_ID | Sentence_No | Sentence
.....................................
a9000006|1|Commercial exploitation over the past two hundred years drove the great Mysticete whales to near extinction.
a9000006|2|Variation in the sizes of populations prior to exploitation, minimal population size during exploitation and current
population sizes permit analyses of the effects of differing levels of exploitation on species with different biogeographical d
istributions and life-history characteristics.
a9000006|3|Dr. Stephen Palumbi at the University of Hawaii will study the genetic population structure of three whale species i
n this context, the Humpback Whale, the Gray Whale and the Bowhead Whale.
a9000006|4|The effect of demographic history will be determined by comparing the genetic structure of the three species.
a9000006|5|Additional studies will be carried out on the Humpback Whale.
a9000006|6|The humpback has a world-wide distribution, but the Atlantic and Pacific populations of the northern hemisphere appe
ar to be discrete populations, as is the population of the southern hemispheric oceans.
a9000006|7|Each of these oceanic populations may be further subdivided into smaller isolates, each with its own migratory patte
rn and somewhat distinct gene pool.
a9000006|8|This study will provide information on the level of genetic isolation among populations and the levels of gene flow
and genealogical relationships among populations.
a9000006|9|This detailed genetic information will facilitate international policy decisions regarding the conservation and mana
gement of these magnificent mammals.
Number of sentences:  9

 Abstract_ID | Sentence_No | Sentence
.....................................
a9000031|1|Studies of chickens have provided serological and nucleic acid probes useful in defining the major histocompatibilit
y complex (MHC) in other avian species.
a9000031|2|Methods used in detecting genetic diversity at loci within the MHC of chickens and mammals will be applied to determ
ining the extent of MHC polymorphism within small populations of ring-necked pheasants, wild turkeys, cranes, Andean condors an
d other species.
a9000031|3|The knowledge and expertise gained from working with the MHC of the chicken should make for rapid progress in defini
ng the polymorphism of the MHC in these species and in detecting the polymorphism of MHC gene pool within small wild and captiv
e populations of these birds.
a9000031|4|Genes within the major histocompatibility complex (MHC) are known to encode molecules that provide the context for r
ecognition of foreign antigens by the immune system.
a9000031|5|Whether a given animal is able to mount an immune response to the challenge of a pathogen is determined, in part, by
the allelic makeup of its MHC.
a9000031|6|In many species, an unusually high degree of polymorphism is maintained at multiple loci within the MHC in freely br
eeding populations.
a9000031|7|The allelic pool within a population presumably provides diversity upon which to draw in the face of environmental c
hallenge.
a9000031|8|The objective of the proposed research is to extend ongoing studies of the MHC of domesticated fowl to include avian
species experiencing severe reduction in population size.
a9000031|9|Knowledge of the MHC gene pool within populations and of the haplotypes of individual animals may be useful in the h
usbandry of species requiring intervention for their preservation.
Number of sentences:  9

 Abstract_ID | Sentence_No | Sentence
.....................................
a9000038|1|This research is part of an on-going program by the principal investigator and associates.
a9000038|2|Topics in the following areas are to be considered: (1) controlled Markov diffusions and nonlinear PDEs; (2) asympto
tic properties of nearly deterministic Markov processes; (3) financial economics applications; (4) singular stochastic control;
(5) computational methods in stochastic control; (6) stochastic calculus of variations; (7) nonlinear estimation.
a9000038|3|Analytical methods based on viscosity solution techniques for nonlinear differential equations as well as probabilis
tic methods will be studied.
a9000038|4|These theoretical studies are the basis for applied problems ranging from decisions at the stock market level to the
control of spaceships.
Number of sentences:  4
```

8. References:

- https://realpython.com/working-with-files-in-python/
- https://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html
- Course content