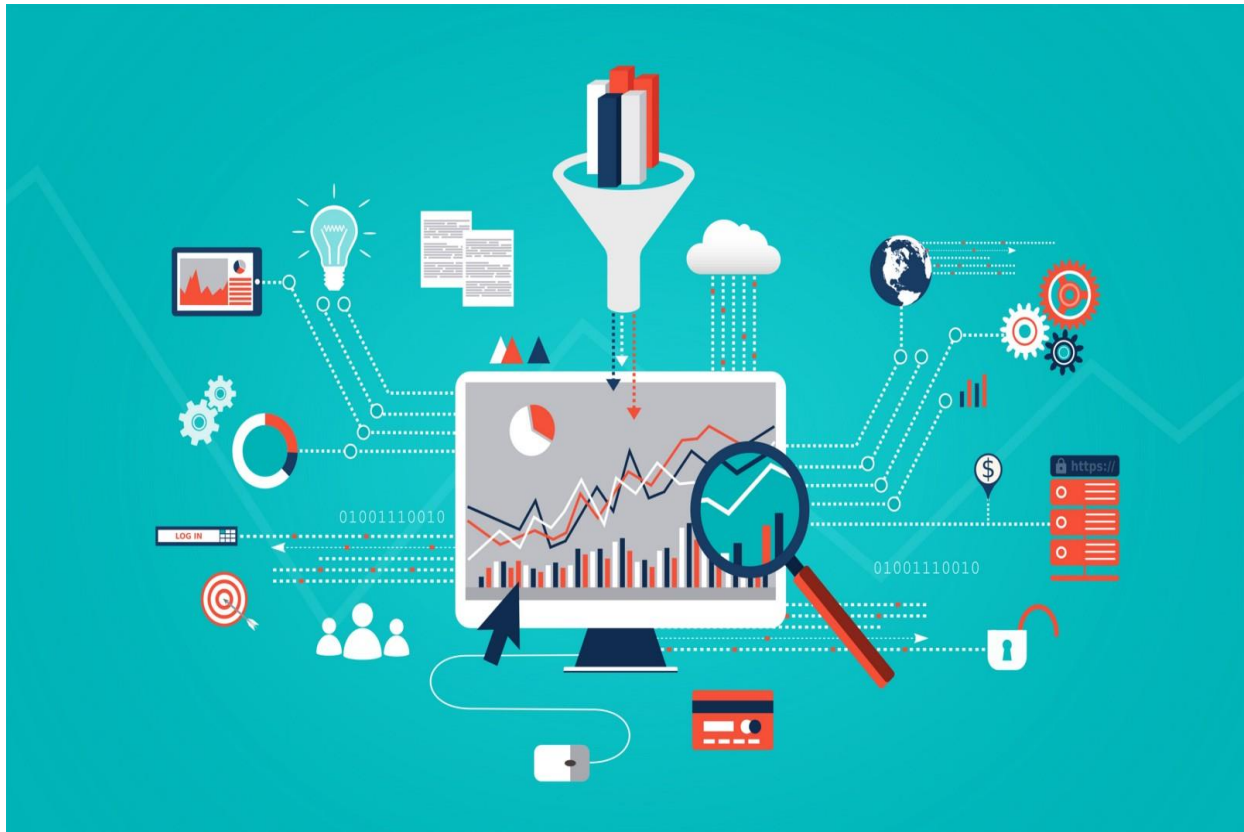


# INFO 5731: Computational Methods for Information Systems

## FINAL PROJECT REPORT

### Natural language processing of behavior analysis citations



**Project Mentor:** Nicole Blank

**Project Leader:** Tejas Kulkarni

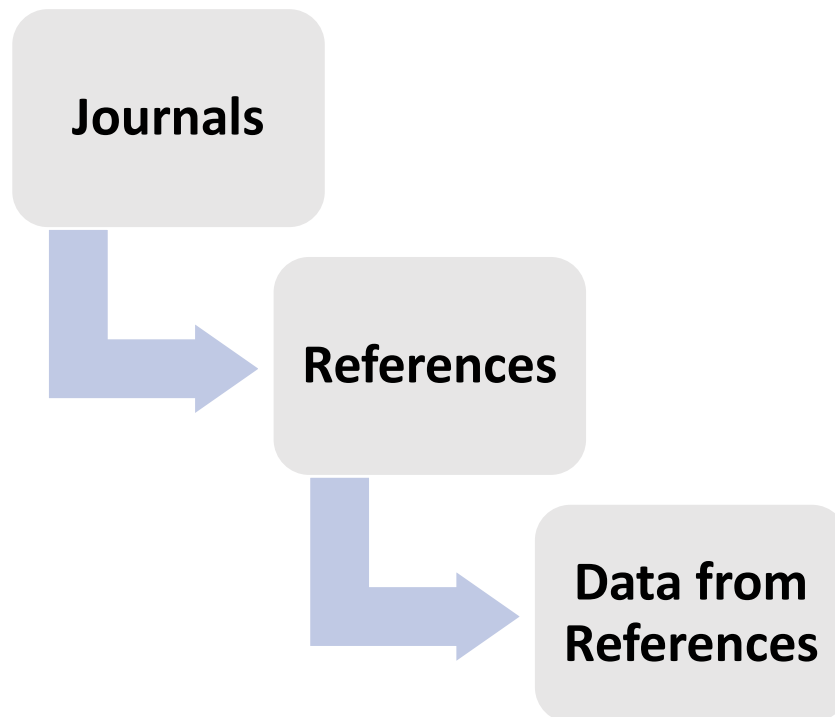
**Project Members:** Bharath Suroju, Anand Pandarapagada, Mayuri Joshi, Shruti Mathur

# Introduction

The key role of behavior analyst is to integrate best available research developments with their work to provide the best services to the client. Across disciplines in which behavior analyst work, there has been consistent gap between what is being researched and implemented. Unfortunately, the limited access to the research development artifacts makes it hard for behavior analyst to work more effectively. This project seeks to gather all these scholarly and scientific artifacts that will improve the credibility of the services offered by behavior analysts. Therefore, having the significant and systematic access to research developments disseminated through formal and informal channels can help to structure their practice settings explicitly in decision making and thereby improving outcomes. This project aims to combine all the references in articles published in nine flagship behavior analytic journals. These journals are mainly related to the knowledge infrastructure of the field of behavior analysis. Gathering all the artifacts in one file would allow us to review and analyze the formal and informal channels that disseminates artifacts. It will also able to identify how many of that references are accessible by behavior analytic practitioners who are not associated with any businesses or organization.

## Data to be Retrieved

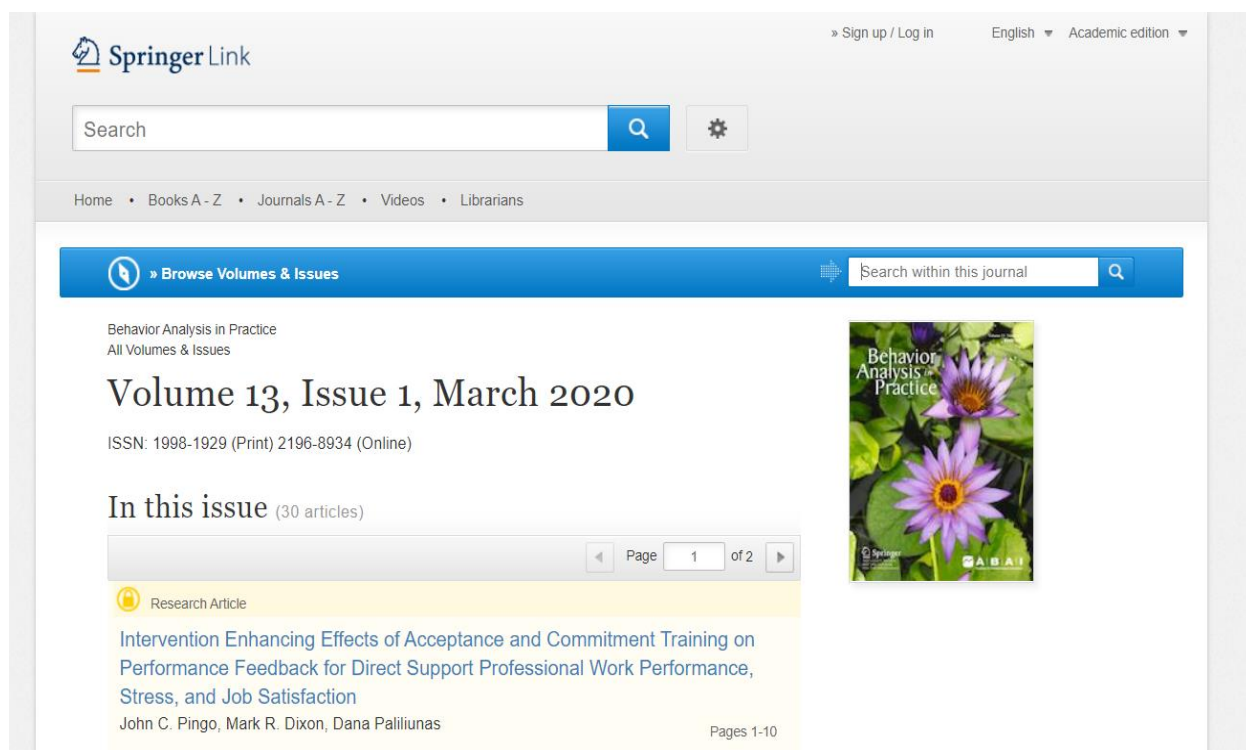
We are extracting references from nine flagship journals that comprises the scope of peer-reviewed journals exclusively publishing behavior analytic research and supported by behavior analytic organizations.



❖ Journal from which References are to be extracted:

- Analysis of Verbal Behavior
- Behavior Analysis in Practice
- Behavior and Social Issues
- European Journal of Behavior Analysis
- Journal of Applied Behavior Analysis
- Journal of the Experimental Analysis of Behavior
- Mexican Journal of Behavior Analysis
- Perspectives on Behavior Science
- Psychological Record

❖ References are extracted from Journals using web scraping and extraction methodologies.



- ❖ Once all the references are scrapped, needful data is gathered from them

<b>Journal Title</b>	
<b>Article Title</b>	
<b>Author Name</b>	
<b>Year in which article is being published</b>	

## Data Retrieval Approaches

Each journal composed of multiple volumes with the number of published issues. The references that need to be extracted, are present in the articles of these issues. The format of the journals' articles varies from one another which leads to follow different methods and packages for extracting the references. Each journal composed of multiple volumes with the number of published issues. The references that need to be extracted, are present in the articles of these issues.



We are using different web scraping methods for extracting references basically web scraping is used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

The format of the journals' articles varies from one another which leads to follow different methods for extracting the references as mentioned below.

#### ❖ Beautiful Soup Package.

Beautiful Soup is the best Library to scrap the data from a particular website or the Internet. And it is most comfortable to work on also. It parses and extracts structured data from HTML and XML documents. Beautiful soup works with urllib. request module. Urllib module defines functions and classes which help in opening URLs (mostly HTTP) in a complex world — basic and digest authentication, redirections, cookies and it opens the url, which can be either a string or a Request object.

#### ❖ Selenium.

Selenium is a framework which is designed to automate test for web applications. With the help of python script, it controls the browser interactions automatically such as link clicks and form submissions. Selenium uses WebDriver which is an open source tool for automated testing of webapps across many browsers. It provides capabilities for navigating to web pages, user input. Chrome Driver is a separate executable that Selenium WebDriver uses to control Chrome. It is maintained by the Chromium team with help from WebDriver contributors.

#### ❖ Regular Expressions. (Regex)

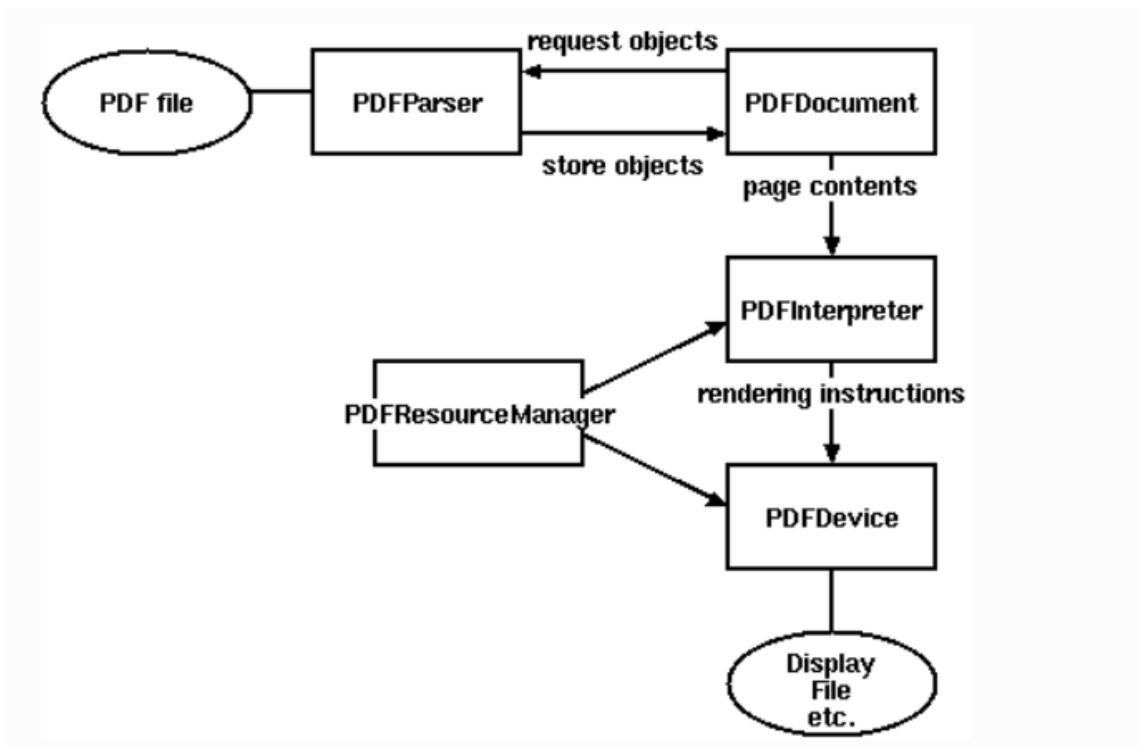
Regular expressions, also called regex is widely used in natural language processing, web applications that require validating string input. A regex pattern is a special language used to represent generic text, numbers or symbols so it can be used to extract texts that conform to that pattern.

There are various parameters which are commonly used in regular expression i.e.

- Pattern: This is the regular expression to be matched.
- String: This is the string, which would be searched to match the pattern at the beginning of string.
- Flags: We can specify different flags using bitwise OR (|). These are modifiers, which are listed in the table below.

## ❖ Pdfminer Tool

PyPDF2 is a tool for extracting information from PDF documents, it focuses entirely on getting and analyzing text data. PyPDF2 allows one to obtain the exact location of text in a page, as well as other information such as fonts or lines. It includes a PDF converter that can transform PDF files into other text formats (such as HTML). It has an extensible PDF parser that can be used for other purposes than text analysis.



PDF files are big and complex structures, parsing a PDF file as a whole is time and memory consuming. However, not every part is needed for most PDF processing tasks. Therefore, PyPDF2 takes a strategy of lazy parsing, which is to parse the stuff only when it's necessary. To parse PDF files, you need to use at least two classes: `PDFParser` and `PDFDocument`. These two objects are associated with each other. `PDFParser` fetches data from a file, and `PDFDocument` stores it. You'll also need `PDFPageInterpreter` to process the page contents and `PDFDevice` to translate it to whatever you need. `PDFResourceManager` is used to store shared resources such as fonts or images.

# Data Retrieval Process and Its Results

## ❖ Process 1

In this process we are extracting references using Beautiful soup, Urllib.request module and Regular expressions.

This process is used to collect references from below Journals –

- Analysis of Verbal Behavior
- Behavior Analysis in Practice
- Behavior and Social Issues
- European Journal of Behavior Analysis

## Results:

```
#getting all the links from the journal webpage in 'wiki' consisting all the volumes and issues and appending to href_list
import urllib.request
from bs4 import BeautifulSoup
import pandas as pd
from urllib.parse import urlparse, urljoin
href_list = []
url = "https://link.springer.com"
wiki = "https://link.springer.com/journal/40617/volumes-and-issues"
href_list = []
#Query the website and return the html to the variable 'page'
page = urllib.request.urlopen(wiki)
soup = BeautifulSoup(page)
for link in soup.find_all("a", class_="u-interface-link u-text-sans-serif u-text-sm"):
    href = link['href']
    href = urljoin(url, href)
    href_list.append(href)
```

```
[ ] #checking whether for a link,next page exists or not if exists append that link also to href_list
def no_of_pages(href_list):
    set1 = set()
    for i in href_list:
        page = urllib.request.urlopen(i)
        soup = BeautifulSoup(page)
        for link in soup.find_all("a", class_="next"):
            href = link['href']
            href2 = url + href
            set1.add(href2)
    return set1

list2 = []
set1 = no_of_pages(href_list)
while(len(set1)>0):
    for value in set1:
        list2.append(value)
    set1 = no_of_pages(list2)
for i in list2:
    href_list.append(i)
```

```

comb = comb + 1
#write or edit the cleaned reference below
comb1 = re.sub(r'^.*[()d{4}.*?[]]\.?', '', str(comb))
journalandarticle = comb1.split('.',1)
#print(journalandarticle)
article = journalandarticle[0]
if len(journalandarticle) != 2 :
    journal = ''
else:
    journal = journalandarticle[1]
    journal = re.sub(r'\s?d.*', '', journal)
authorandYear = re.findall(r'.*[()d{4}.*?[]]', str(comb))
year = re.findall(r'd{4}', str(authorandYear))
if len(year)<1:
    year = ''
else:
    year = year[0]
if len(authorandYear)<1:
    authorandYear = ''
else:
    authorandYear = authorandYear[0]
author = re.sub(r'[()d{4}.*?[]]', '', authorandYear)
print(author)
sheet.write(row,0,str(journal))
sheet.write(row,1,str(article))
sheet.write(row,2,str(year))
sheet.write(row,3,str(author))
row+=1
comb = ''

```

Journal	Article	Year	Author
Journal of Behavioral Education	Teaching non-target information to children with disabilities: An examination of the instructive feedback literature	2019	Albarran, S. A., & Sandbank, M. P.
Journal of Experimental Child Psychology	Reinforcement control of generalized imitation in young children	1964	Baer, D. M., & Sherman, J. A.
). Hoboken, NJ: Wiley-Blackwell.	Understanding behaviorism: Behavior, culture, evolution (2nd ed	2005	Baum, W. M.
Analysis of Verbal Behavior	Using instructive feedback to increase response variability during intraverbal training for children with autism spectrum disorder	2015	Carroll, R. A., & Kodak, T.
The Analysis of Verbal Behavior	The effects of blocking and joint control training on sequencing visual stimuli	2016	Clough, C. W., Meyer, C. S., & Miguel, C. F.
The Journal of Special Education	Use of constant time delay in small group instruction: A study of observational and incidental learning	1990	Doyle, P. M., Gast, D. L., Wolery, M., & Ault, M. J.



## ❖ Process 2

In this process we are extracting references using PDFMiner i.e. using PDFPageInterpreter, text converter, PDFResourceManager.

This process is used to collect references from below Journal –

- Mexican Journal of Behavior Analysis

### Results:

```
string = ''
list_of_files = glob.glob(r'C:/bharath/5731/some/Data2/Data2/*.pdf')
S = open('file2.txt',mode = 'w',encoding="utf-8")
for file in list_of_files:
    wholeFile = convert_pdf_to_txt(file)
    # Using ^Referenc.* to accommodate the word References (English) or Referencias (Spanish)
    match = re.search(r'^Referenc.*', wholeFile, re.MULTILINE|re.DOTALL)
    if match is None:
        pass
    else:
        string = string + match.group(0) + '\n'
# writing to a text file
for line in string:
    S.write(line)
S.close()
```

```
string = ''
list_of_files = glob.glob(r'C:/bharath/5731/some/Data2/Data2/*.pdf')
S = open('file2.txt',mode = 'w',encoding="utf-8")
for file in list_of_files:
    wholeFile = convert_pdf_to_txt(file)
    # Using ^Referenc.* to accommodate the word References (English) or Referencias (Spanish)
    match = re.search(r'^Referenc.*', wholeFile, re.MULTILINE|re.DOTALL)
    if match is None:
        pass
    else:
        string = string + match.group(0) + '\n'
# writing to a text file
for line in string:
    S.write(line)
S.close()
```

Journal	Article	Year	Author
Behavioral and Brain Sciences	Unifying the Behavioral Sciences	2007	Gintis, H.
Behavioral and Brain Sciences	Unifying the Behavioral Sciences	2007	Gintis, H.
Revista Mexicana deAnálisis de la Conducta	Efectos del intervalo respuesta-reforzador y del ciclode reforzamiento en un programa de demora variable	1989	Ávila, R. & Bruner, C. A.

### ❖ Process 3

In this process we are extracting references using Selenium from below Journals.

- Journal of Applied Behavior Analysis
- Journal of the Experimental Analysis of Behavior
- Perspectives on Behavior Science
- Psychological Record

#### Results:

```
!apt-get update
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver /usr/bin
!pip install selenium
import os
import re,datetime
import time
import csv
from selenium.common.exceptions import NoSuchElementException
from google.colab import files
from selenium import webdriver
```

```
from selenium import webdriver

class In_Class_Assignment():
    def get_data(self,list_of_journal):
        list_of_data=[]
        print(list_of_journal)
        obj = In_Class_Assignment()
        # Launch chrome browser.
        options = webdriver.ChromeOptions()
        # Runs browser in background.
        options.add_argument('--headless')
        options.add_argument('--no-sandbox')
        options.add_argument('--disable-dev-shm-usage')
        options.add_argument('--disable-gpu')
        options.add_argument("--window-size=1920,1080")
        options.add_argument("start-maximized")
        options.add_argument("disable-infobars")
        options.add_argument("--disable-extensions")
        driver = webdriver.Chrome('chromedriver',options=options)
```

```

list_of_data = list_of_journal[journal]
print(list_of_data)
print(list_of_data[2])
#driver.get is use to access url.
driver.get(list_of_data[2])
try:
    if driver.find_element_by_xpath("//button[@aria-expanded='false'])[8]").is_displayed():
        # finds year,but before that it checks it is already expanded or not.
        driver.find_element_by_xpath("//div[text()='Date Published']").click()
        time.sleep(5)
except NoSuchElementException as msg:
    print("No such element found")
driver.find_element_by_xpath("//input[@id='start-year']").clear()
driver.find_element_by_xpath("//input[@id='start-year']").send_keys(list_of_data[0])
driver.find_element_by_xpath("//input[@id='end-year']").clear()
driver.find_element_by_xpath("//input[@id='end-year']").send_keys(list_of_data[1])
driver.find_element_by_xpath("//input[@id='end-year']/following::input[@type='submit'])[1]").click()
time.sleep(10)
#It find how many page are there on which articles are present.
total_pages = driver.find_element_by_xpath("//span[@class='number-of-pages'])[1]").text
count = int(total_pages)

# Actual content i.e. data which we want to extract from webpage is saved in reference_text in
reference_text = list_of_ref[j].text
author = obj.get_author(reference_text)
if author is None:
    continue
author.append(list_of_data[3])
obj.writeintocsv(author)
driver.back()
else:
    driver.back()
button=driver.find_element_by_xpath("//img[contains(@src,'/images/arrow-right.png')])[1]")
driver.execute_script("arguments[0].scrollIntoView();", button)
button.click()
time.sleep(10)

```

Journal Names	Article Title	Year	Author
Perspectives on Behavior Science	Performance feedback and probabilistic bonus contingencies among employees in a human service organization	2005	Cook, T., & Dixon, M. R.
Perspectives on Behavior Science	Quick Wins! Accelerating School Transformation through Science, Engagement, and Leadership	2016	Gavoni, P., & Rodriguez, M.
Perspectives on Behavior Science	Graphic feedback, performance feedback, and goal setting increased staff compliance with a data collection task at a large residential facility	2016	Gil, P. J., & Carter, S. L.

# Data Analysis

Data analysis is a process of inspecting, cleansing, transforming and modeling data to discover useful information, informing conclusion and supporting decision-making. The results of the data analysis in this project will provide multiple facets and approaches that will help to make the decisions more scientific. Each journal composed of multiple volumes with the number of published issues. The references that need to be extracted, are present in the articles of these issues. The format of the journals' articles varies from one another which leads to follow different methods for extracting the references as mentioned below.

- Article Topic Modeling.
- Journals that have appeared frequently.
- Articles name that have appeared frequently.
- Authors name that have appeared frequently.
- Year in which most of the articles are published

For conducting above analysis, we have used LDA, Pandas library.

## ❖ Latent Dirichlet Allocation (LDA)

LDA is a topic model that generates topics based on word frequency from a set of documents. It is particularly useful for finding reasonably accurate mixtures of topics within a given document set. Topic modeling tries to group the documents into clusters based on similar characteristics. A typical example of topic modeling is clustering a large number of newspaper articles that belong to the same category. To generate an LDA model, we need to understand how frequently each term occurs within each document. To do that, we need to construct a document-term matrix with a package called genism.

- What LDA actual does.?
- Determine the number of words in a document
- Determine the mixture of topics in that document
- Using each topic's multinomial distribution, output words to fill the document's word slot  
–i.e. it works on probability percentage and it fills that word according to that.

## ❖ Pandas

Pandas is a library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It uses data frames. A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas Data Frame consists of three principal components, the data, rows, and columns.

## Data Analysis Results

- Article Topic Modeling.

```
M texts=[]
from nltk.tokenize import RegexpTokenizer
from gensim import corpora, models
import gensim
tokenizer = RegexpTokenizer(r'\w+')
for i in pd.Series(df['Journal']):
    tokens = tokenizer.tokenize(i)
    texts.append(tokens)
dictionary = corpora.Dictionary(texts)
corpus = [dictionary.doc2bow(text) for text in texts]
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =5 ,id2word = dictionary,passes =20)

M print(ldamodel.print_topics(num_topics=5,num_words =5))

[(0, '0.102*"new" + 0.094*"york" + 0.044*"press" + 0.029*"science" + 0.027*"university"', (1, '0.048*"social" + 0.039*"eff
ects" + 0.032*"n" + 0.029*"personality" + 0.024*"1967"', (2, '0.073*"amp" + 0.047*"j" + 0.026*"r" + 0.021*"h" + 0.021*"psy
chol"', (3, '0.100*"psychological" + 0.072*"behavior" + 0.041*"record" + 0.033*"learning" + 0.031*"research"', (4, '0.194
*"journal" + 0.138*"behavior" + 0.107*"experimental" + 0.103*"analysis" + 0.093*"psychology"')]
```

- Journals that have appeared frequently.

journal experimental analysis behavior	27189
journal applied behavior analysis	11009
psychological record	10030
behavior analyst	7962
american psychologist	4487
behavioural processes	3927
psychological review	3802
science	3384
journal experimental psychology	3046
revista mexicana de análisis de la conducta	2996
Name: Journal, dtype: int64	

- Articles name that have appeared frequently.

science human behavior	1886
verbal behavior	1716
behaviorism	1654
psychology behaviorist views	1216
b	1190
john b	972
struggle scientific authority reception watsons behaviorism 19131920	815
na	798
resurgence previously reinforced responding research application	782
origins behaviorism american psychology 18701920	780
Name: Article, dtype: int64	

- Authors name that have appeared frequently.

```
freq = pd.Series(df['Year']).value_counts()[:10]
freq
```


```
2009    13153
2008    11468
2001    10969
2011    10902
2000    10489
2010    10408
2003    10295
2006     9905
2002     9628
2013     9317
Name: Year, dtype: int64
```

---

- Year in which most of the articles are published

```
[ ] freq = pd.Series(df['Author']).value_counts()[:10]
freq
```

```

 Skinner, B. F.    11453
Watson, J. B.     6913
Epstein, R.      2735
Ribes, E.        2178
Sidman, M.       2047
Rachlin, H.      1678
Samelson, F.     1434
Kantor, J. R.    1403
Dewsbury, D. A.  1368
Catania, A. C.   1283
Name: Author, dtype: int64
```

## Conclusion

- We have collected references from all articles of nine flagship by scraping web pages of Journals published on [springer.com](http://springer.com) and [onlinelibrary.wiley.com](http://onlinelibrary.wiley.com) using beautiful soup package, pdfminer, Selenium and Regular expressions.
- All the extracted references are then saved in CSV file and data analysis is performed and number of frequently appeared articles name are calculated.
- Depending on analysis, now dissertation can be being applied to delineate the formal and informal dissemination artifacts and identified how many are accessible by these behaviors.

## Repository Link for Code and Result



[https://github.com/tejask666/Tejas\\_INFO5731\\_Spring2020/tree/master/tejask666/Tejas\\_INFO5731\\_Spring2020/INFO%205731\\_Project\\_Team\\_Project](https://github.com/tejask666/Tejas_INFO5731_Spring2020/tree/master/tejask666/Tejas_INFO5731_Spring2020/INFO%205731_Project_Team_Project)

# References

- Dhankhad, S. (2019, April 01). Forget APIs Do Python Scraping Using Beautiful Soup, Import Data File from the web: Part 2. Retrieved May 04, 2020, from <https://towardsdatascience.com/forget-apis-do-python-scraping-using-beautiful-soup-import-data-file-from-the-web-part-2-27af5d666246>
- Huang, H. (2019, June 18). Quick Web Scraping with Python & Beautiful Soup. Retrieved May 04, 2020, from <https://levelup.gitconnected.com/quick-web-scraping-with-python-beautiful-soup-4dde18468f1f>
- Pocs, M. (2020, February 20). Web Scraping with Selenium in Python. Retrieved May 04, 2020, from <https://levelup.gitconnected.com/web-scraping-with-selenium-in-python-8fde2f0fd559>
- Python - Regular Expressions. (n.d.). Retrieved May 05, 2020, from [https://www.tutorialspoint.com/python/python\\_reg\\_expressions.htm](https://www.tutorialspoint.com/python/python_reg_expressions.htm)