| **Hypergraph Plot** |
| --- |

**Follow the directions listed below to make a Hypergraph:**

Part 1: Upload the Text to Analyze

1. Go to [Wolfram Cloud](https://www.wolframcloud.com) and login (use old or make free account)
2. Make a new folder: click “New” then “Folder” in the corner of your dashboard
3. Name your folder “Merged Files”. This is where you’ll put your text files
4. Make a new text file: click “New” then “Text” in the corner of your dashboard
5. Name your text file whatever you’d like
6. Put this text file in the “Merged Files” folder
7. Copy and paste the text you want to analyze in the text file
8. Verify that it has been saved by reopening the file. Sometimes it does not save.

Part 2: Making the Code

1. Make a new notebook: click “New” then “Notebook” in the corner of your dashboard
2. Name your notebook
3. In the left corner, hover your mouse until a plus + icon appears, then click the option titled “Wolfram Language Input”
4. Paste into this box the code below
5. Change what is highlighted in blue to fit your particular application (more details below)

| article = Import["MergedFiles/**TEXT FILE NAME.txt**"];  article = ToLowerCase[article];  article = StringDelete[article, "http://" ~~ (LetterCharacter | DigitCharacter | ".") ..];  article = StringDelete[article, (DigitCharacter) ..];  article = StringReplace[article, PunctuationCharacter -> " "];  topN = 25 **DEFAULT IS 25 KEYWORDS, CHANGE IF YOU WANT MORE/LESS**;  words = {};  words = TextWords[article];  Pos = ("a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z");  indices = Position[words, Pos];  words = Delete[words, indices];  words = DeleteStopwords[words];  words = words /. {"ca" -> "calcium", "ca2" -> "calcium", "cell" -> "cells", "proteins" -> "protein"};  words = Reverse[SortBy[Tally[words], Last]];  For[i = 1, i <= topN, i++, words[[i]] = First[words[[i]]]];  words = words[[1 ;; topN]];  words  maxN = 3;  broadTerms = {};  words1 = {};  words2 = {};  For[i = 1, i <= topN, i++,If[Length[Position[WordData[], Extract[words, i]]] > 0,  broadTerms = Append[broadTerms, Flatten[Values[WordData[Extract[words, i], "BroaderTerms"]]]];  words1 = Append[words1, Extract[words, i]], words2 = Append[words2, Extract[words, i]]]];  For[i = 1, i <= Length[broadTerms], i++, If[Length[broadTerms[[i]]] > maxN, broadTerms[[i]] = broadTerms[[i]][[1 ;; maxN]]]];  edgeList = {};  edgeList = Append[edgeList,Prepend[words1, "Chapter 1.1.3"]] ;  For[i = 1, i <= Length[words1], i++, edgeList = Append[edgeList, Append[broadTerms[[i]], words1[[i]]]]];  ResourceFunction["HypergraphPlot"][edgeList, VertexSize -> 0.01, VertexLabels -> Automatic, PlotTheme -> "Detailed", SubsetEdge -> False, SubsetBoundaryScale -> 7, VertexStyle ->  Red,"BaseLayout" -> "RadialEmbedding","SubsetEdgeStyle" -> ColorData[100, "ColorList"]] |
| --- |

| **Tree Diagram Plot** |
| --- |

**Follow the directions listed below to make a Tree Diagram:**

Part 1: Upload the Text to Analyze

1. Go to [Wolfram Cloud](https://www.wolframcloud.com) and login (use old or make free account)
2. Make a new folder: click “New” then “Folder” in the corner of your dashboard
3. Name your folder “Merged Files”. This is where you’ll put your text files
4. Make a new text file: click “New” then “Text” in the corner of your dashboard
5. Name your text file whatever you’d like
6. Put this text file in the “Merged Files” folder
7. Copy and paste the text you want to analyze in the text file
8. Verify that it has been saved by reopening the file. Sometimes it does not save.

Part 2: Making the Code

1. Make a new notebook: click “New” then “Notebook” in the corner of your dashboard
2. Name your notebook
3. In the left corner, hover your mouse until a plus + icon appears, then click the option titled “Wolfram Language Input”
4. Paste into this box the code below
5. Change what is highlighted in blue to fit your particular application (more details below)

| article = Import["MergedFiles/**TEXT FILE NAME.txt**"];  article = ToLowerCase[article];  article = StringDelete[article, "http://" ~~ (LetterCharacter | DigitCharacter | ".") ..];  article = StringDelete[article, (DigitCharacter) ..];  article = StringReplace[article, PunctuationCharacter -> " "];  topN = 25 **DEFAULT IS 25 KEYWORDS, CHANGE IF YOU WANT MORE/LESS**;  words = {};  words = TextWords[article];  Pos = ("a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z");  indices = Position[words, Pos];  words = Delete[words, indices];  words = DeleteStopwords[words];  words  words = Reverse[SortBy[Tally[words], Last]];  For[i = 1, i <= topN, i++, words[[i]] = First[words[[i]]]];  words = words[[1 ;; topN]];  words  maxN = 3;  broadTerms = {};  words1 = {};  words2 = {};  For[i = 1, i <= topN, i++,If[Length[Position[WordData[], Extract[words, i]]] > 0,  broadTerms = Append[broadTerms, Flatten[Values[WordData[Extract[words, i], "BroaderTerms"]]]];  words1 = Append[words1, Extract[words, i]], words2 = Append[words2, Extract[words, i]]]];  For[i = 1, i <= Length[broadTerms], i++, If[Length[broadTerms[[i]]] > maxN, broadTerms[[i]] = broadTerms[[i]][[1 ;; maxN]]]];  edgeList = {};  For[i = 1 , i <= Length[broadTerms], i++, edgeList = Append[edgeList, broadTerms[[i]]]];  For[i = 1 , i <= Length[broadTerms], i++, For[j = 1, j <= Length[edgeList[[i]]], j++, edgeList[[i]][[j]] = words1[[i]] -> edgeList[[i]][[j]]]];  For[i = 1 , i <= Length[words1], i++, edgeList = Append[edgeList, {"**GIVE YOUR TREE DIAGRAM A NAME**" -> words1[[i]]}]];  TreePlot[Flatten[edgeList], VertexSize -> 0.1, VertexLabels -> Automatic] |
| --- |

| **Bar Diagrams** |
| --- |

**Follow the directions listed below to make a Bar Diagram:**

Part 1: Upload the Text to Analyze

1. Go to Google Colab here: <https://colab.research.google.com>
2. Click on the folder icon on the left side of the screen
3. Wait for the files in this panel to load- one of the folders is called “sample-data”
4. Create a new folder inside this “sample-data” folder. Name this new folder “Data”.
5. Upload your .txt files into this “Data” Folder
6. Rename each .txt file as a number corresponding to the article number it is in that section. For example, if you have 13 articles in a section, name the first one “1.txt”, then “2.txt”, “3.txt”, etc.

Part 2: Making the Code

1. Add a new cell to put code in
2. Copy and Paste the following code shown below, making edits as needed
3. Run the code in by clicking the play button on the right side of the cell screen

| !pip install selenium  from matplotlib import pyplot as plt  from collections import Counter  import nltk  nltk.download('stopwords')  from nltk.corpus import stopwords  from nltk.tokenize import word\_tokenize  # ------------------------ VARIABLES ------------------------ #  articles = []  other\_stop\_words = [“”, “”, “”, **“ADD WORDS YOU WANT TO DELETE FROM GRAPHS BY PUTTING THEM HERE IN QUOTES”**]  # ------------------------ READ FILES ------------------------ #  for article in range(**HOW MANY FILES ANALYZING**):  article\_number = str(article + 1)  path = '/content/sample\_data/Data/'+ article\_number + '.txt'  with open(path, encoding ='ISO-8859-1') as fopen:  article = fopen.read()  articles.append(article)  # ------------------------ FREQUENCY ------------------------ #  def frequency(str):  text = str.lower().split()  dictionary = {}  # add all words to the dictionary  for word in text:  if word not in dictionary:  dictionary[word] = 1  else:  number = dictionary.get(word) + 1  dictionary.update({word : number})  stop\_words = set(stopwords.words('english'))  delete = [key for key in dictionary if key in stop\_words or key in other\_stop\_words]  for key in delete: del dictionary[key]  dictionary = dict(Counter(dictionary).most\_common(10))  print(dictionary)  return dictionary  # ------------------------ PLOTTING ------------------------ #  article = 1  for i in range(**HOW MANY FILES ANALYZING**):  fig, axs = ig, ax = plt.subplots(figsize=(25, 5))  dictionary = frequency(articles[article - 1])  # sort items by largest frequency  dictionary = dict(sorted(dictionary.items(), key = lambda kv: kv[1], reverse = True))  title = "**ARTICLE TITLE NUMBER** ", str(article), " Bar Graph: Word Frequency"  keys = dictionary.keys()  values = dictionary.values()  axs.set\_title(title)  axs.bar(keys, values, color = (0.75,0.5,0.2,0.9))  article = article + 1  plt.show() |
| --- |