In [1]:
```
#Importing relevant libraries
from pdfminer.high_level import extract_text
import PyPDF2
from PyPDF2 import PdfReader
import re
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk import download
from gensim import corpora, models
from gensim.models import CoherenceModel
import os
import statistics
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.stats import pearsonr
import matplotlib.pyplot
import seaborn as sns
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from gensim.models.coherencemodel import CoherenceModel
import tensorflow as tf
import os
import pdfplumber
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import re
from gensim.corpora import Dictionary
from gensim.models import HdpModel
```

In [2]:
```
# No preprocessing is needed for BERTopic
# Just keep the text in its original form

# Create a dictionary from the preprocessed text
# BERTopic does not require a dictionary or bag-of-words representation

# Create a corpus
# BERTopic does not require a corpus with bag-of-words representation
# Instead, BERTopic directly works with the text data to generate embeddings

# texts variable now contains the raw text from the PDFs, which can be used di

# Directory path containing PDF files
pdf_directory = '/Users/cdlacey/TMU_DataScience/CIND820/Dataset_BERTopic'

# List all PDF files in the directory
pdf_files = [os.path.join(pdf_directory, file) for file in os.listdir(pdf_dire

texts = []

# Loop through each PDF file and extract text
for pdf_file in pdf_files:
    with pdfplumber.open(pdf_file) as pdf:
        text = ""
```

```
        for page in pdf.pages:
            text += page.extract_text()
        texts.append(text)
```

In [ ]:
```
import transformers
from bertopic import BERTopic

# Train the BERTopic model
bertopic_model = BERTopic()
topics, _ = bertopic_model.fit_transform(texts)
```

In [ ]:
```
from sentence_transformers import SentenceTransformer
from bertopic import BERTopic

# Step 1: Convert Texts to Sentence Embeddings
# Load a pre-trained sentence transformer model
model = SentenceTransformer('paraphrase-MiniLM-L6-v2')

# Generate sentence embeddings for each document
sentence_embeddings = model.encode(preprocessed_texts)

# Step 2: Fit BERTopic Model
# Initialize BERTopic model
bertopic_model = BERTopic()

# Fit BERTopic model to the sentence embeddings
topics, _ = bertopic_model.fit_transform(sentence_embeddings)
```

In [ ]:
```

```