Kaushik Dutta, PhD

Daniel Clinton, PhD

Suryateja Chalapati
(U3699-1670)

ISM6905 – Growth Prediction from Emerging Technologies Based on SEC 10K & YouTube Data

Independent Study Report

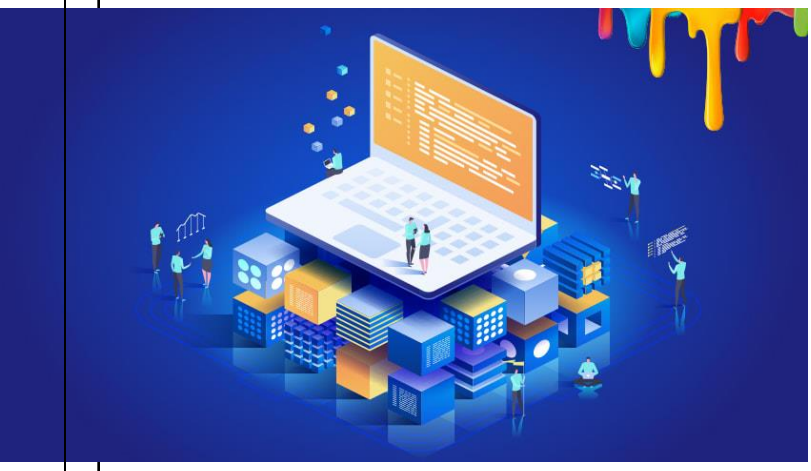

TABLE OF CONTENTS

1. INTRODUCTION | Overview

1.1. Project Overview

Summary: The topic of the independent research project is to predict Emerging technologies from the Gartner reports based on the companies' financials like 10K reports and from platform like YouTube where information is being shared form one node to another.

1.2. To-Do

My Tasks: This is a very big project with a lot of scope in predictive text analytics where we used topic modelling techniques like TF-IDF, LDA, LSA etc. to gain valuable insights from the data. My tasks were specifically related to gathering data which is done in the following steps:

1. Extract SEC 10K data for companies in technology domain by web scraping and storing the Business and Risk sections of the reports as a text corpus.
2. Then, extract YouTube transcripts based on emerging technologies by Gartner reports for the last 6 years and store as another text corpus.
3. Then perform topic modelling on the corpus and apply TF-IDF, LDA and LSA on the data.

I did the corpus extraction for both SEC 10k and YouTube data which is done by web scraping using python and R studio. Also involved with organizing and storing the data year wise and performing sanity checks to make sure we have the corpus available as intended.

Technical Requirements Road Map:

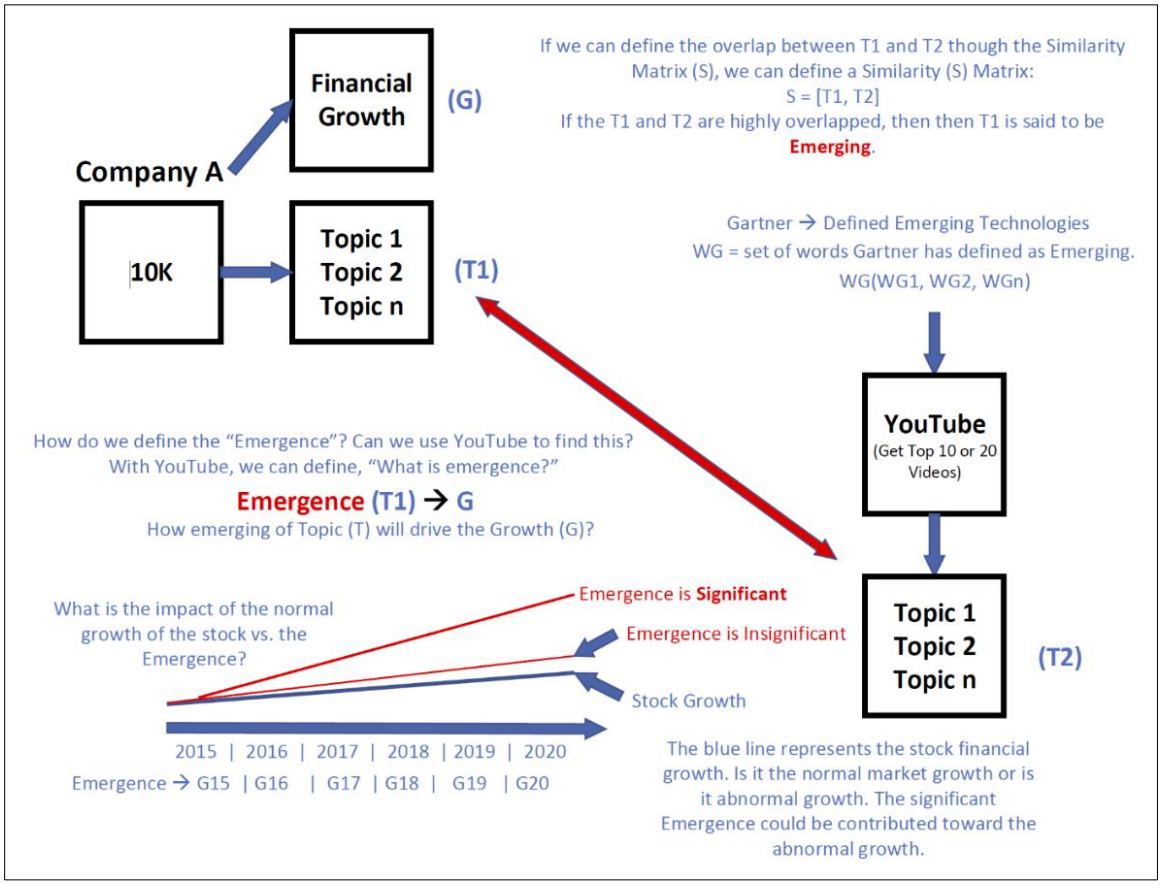

Fig. 1: Flow Chart

1.3. Project Hierarchy

The project hierarchy as follows:

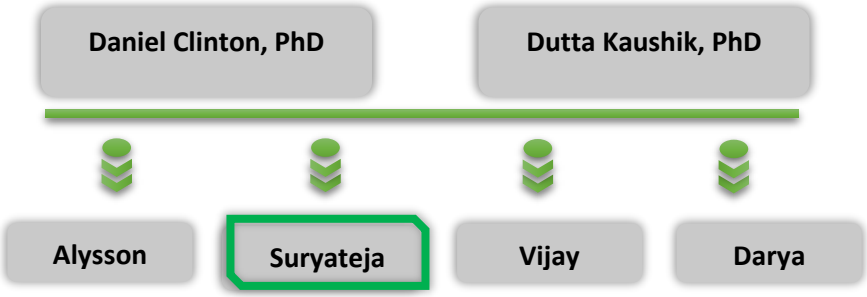

Fig. 2: Hierarchy

1.4. Literature Review

The principal goal of our literature review is to identify specific sources of financial and conversational text data which can then be mined for insights. For this study we chose SEC 10K data for all the businesses that fall in the Information technology domain. For the growth estimation aspect, we have chosen YouTube as data sourcing could be done via the API. The key emergence technology terms were sourced from Gartner's Hype cycle for emerging technologies reports from the past six years (2015-2020).

2. PROGRESS | To-Do

2.1. YouTube Corpus

This script extracts individual transcripts for about 100-130 videos on all years and saves as a text file.

```
def getpage(a):
    nextpage=a
    return nextpage

def getid(m):
    ext=[]
    for i in range(0,len(m)):
        ext.append(m[i]['id'] ['videoId'])
    return ext
def geturl(b):
    p="pageToken="+ b+"&"
    newurl=part0+part1+p+part2+part3+begin_year+part4+end_year+part5+part6
    return newurl


keywords = ["Augmented Data"]

for keyword in keywords:

    ext=[]
    lst1=[]

    userquery=keyword
    get_begin_year='2017'
    get_end_year='2018'
    part0="https://www.googleapis.com/youtube/v3/search?"
    part1="part=snippet&maxResults=25&"
    part2="q="+ str(userquery)
    part3="&type=video&videoCaption=closedCaption&publishedAfter="
    begin_year=str(get_begin_year)
    part4="-01-01T00:00:00Z&publishedBefore="
    end_year=str(get_end_year)
    part5="-01-01T00:00:00Z&fields=items(id(channelId%2CvideoId))%2CnextPageToken%2CprevPageToken"
    # Change this line with your YouTube Data API Key
    part6="&key=AIzaSyBEZ73MUbzufi-L6K5P9eJkJbu8-bcagPA"

    # Surya: AIzaSyBEZ73MUbzufi-L6K5P9eJkJbu8-bcagPA
    # Darya: AIzaSyBSUE45QS_0OdiWvvgmUIcs6nTiGmcwuRo
    URL=str(part0+part1+part2+part3+begin_year+part4+end_year+part5+part6)
    page = requests.get(URL)

    mystring=str(page.content.decode("utf-8"))
    mydict = json.loads(mystring)
    ext=getid(mydict['items'])
    lst1=lst1+ext
```

Fig. 3: YouTube Scraper

2.2. SEC 10K Corpus

This script extracts the business and risk sections of 10K reports for all the selected business for last 10 years and saves those as text files.

```
In [1]: # importing required libraries
import re
import os
import time
import urllib.request
import requests
import unicodedata
import numpy as np
import pandas as pd
from bs4 import BeautifulSoup

In [2]: # pip install selenium --user
# conda install -c anaconda beautifulsoup4

In [3]: def dfSec10k(cik_inp):
    # set the central index key
    cik = cik_inp

    # pass the main url from SEC
    url = 'https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=' + cik + '&type=10-k&dateb=&owner=exclude&cou

    # grab the html
    try:
        page = urllib.request.urlopen(url)
    except:
        print("An error occured.")

    soup = BeautifulSoup(page, "html.parser")

    list_td = []
    for item in soup.find_all('td',class_=["small",""]):
        list_td.append(item.text)

    list_td_2 = []
    for item1 in soup.find_all('td',class_=["small",""]):
        for i in item1.find_next('td'):
            list_td_2.append(i)

    df = pd.DataFrame(list_td)
    df1 = pd.DataFrame(list_td_2)

    df.columns = ['table']
    df1.columns = ['date']

    result = pd.concat([df, df1], axis=1)

    # Data cleanup step - 1
    result['date'] = result['date'].astype(str)
    result['date'] = pd.to_datetime(result['date'])
    result['year'] = pd.DatetimeIndex(result['date']).year
    del result['date']
```

Fig. 4: SEC 10K Scraper

This script organizes all the text files into groups for all the selected businesses. It also removes all the files that are either duplicates or has no data.

```
In [ ]: import os
import gensim
import pandas as pd
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
from nltk.stem import WordNetLemmatizer, SnowballStemmer
from nltk.stem.porter import *
import numpy as np
np.random.seed(2021)
import nltk
nltk.download('wordnet')

# Get all file names in a folder
os.chdir('C:/Users/surya/Downloads/Career/M.S. [2020-22]/3.Spring_2021/[ISM6905.001S21.12437 Independent Study]/Temp_Fil
path = os.getcwd()
os.chdir('C:/Users/surya/Downloads/Career/M.S. [2020-22]/3.Spring_2021/[ISM6905.001S21.12437 Independent Study]/')
files = os.listdir(path)
files

# Filter only .xlsx files from the list
files_xls = [f for f in files if f[-4:] == 'xlsx']
files_xls

# Create a dataframe and append data from excel files to the dataframe
# for f in files_xls:
#     data = pd.read_excel('C:/Users/surya/Downloads/Career/M.S. [2020-22]/3.Spring_2021/[ISM6905.001S21.12437 Independe
#     df = df.append(data)
```

Fig. 6: Files Organizer

2.3. Topic Modelling - LDA

This script generates the topic models on the corpus for both YouTube and SEC 10K data.

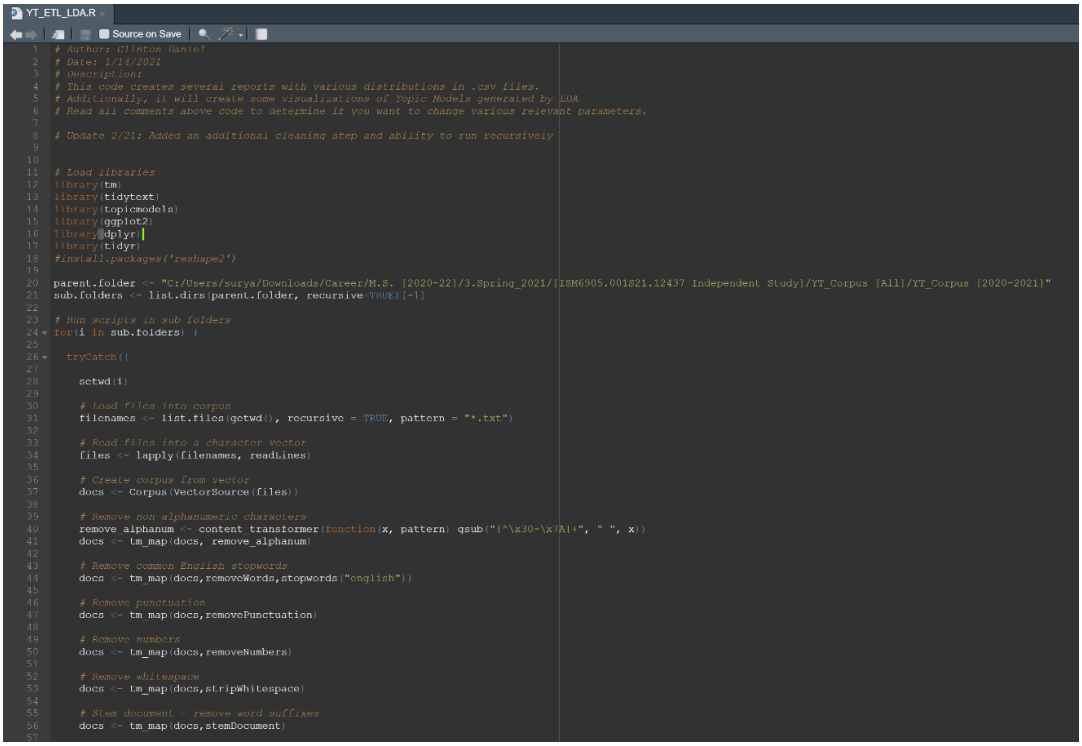

Fig. 7: PHP Randomizer

3. CONCLUSION | Beyond Learning

Overall, this Independent study was a very good learning experience. I have learnt a lot about web-scraping and advanced python. Since 10K reports are very unorganized I had to find workarounds to scrape the data such that we do not lose any valuable information. I have also learned how to use some of the advanced libraries in R for data preprocessing. I learned about Topic Modelling and specifically about Latent Semantic Analysis (LSA) and Singular Value Decomposition (SVD), Latent Dirichlet Allocation (LDA), Cosine Similarity and how these techniques are used to make predictive text analytics. The next steps in the project are to apply statistical methods to predict the correlation between both the sources (10K and YouTube) on the emerging technologies and how these trends can be used to estimate the emergence.

4. REFERENCES | Credits

Following is the list of all the references:

- https://github.com/cedanie2/ism6905-Spring2021
- https://www.sec.gov/edgar/searchedgar/companysearch.html
- https://developers.google.com/youtube/v3

----------[End] ----------