# DEPARTMENT OF COMPUTER SCIENCE
# UNIVERSITY AT BUFFALO

## CSE 587

## LAB 2

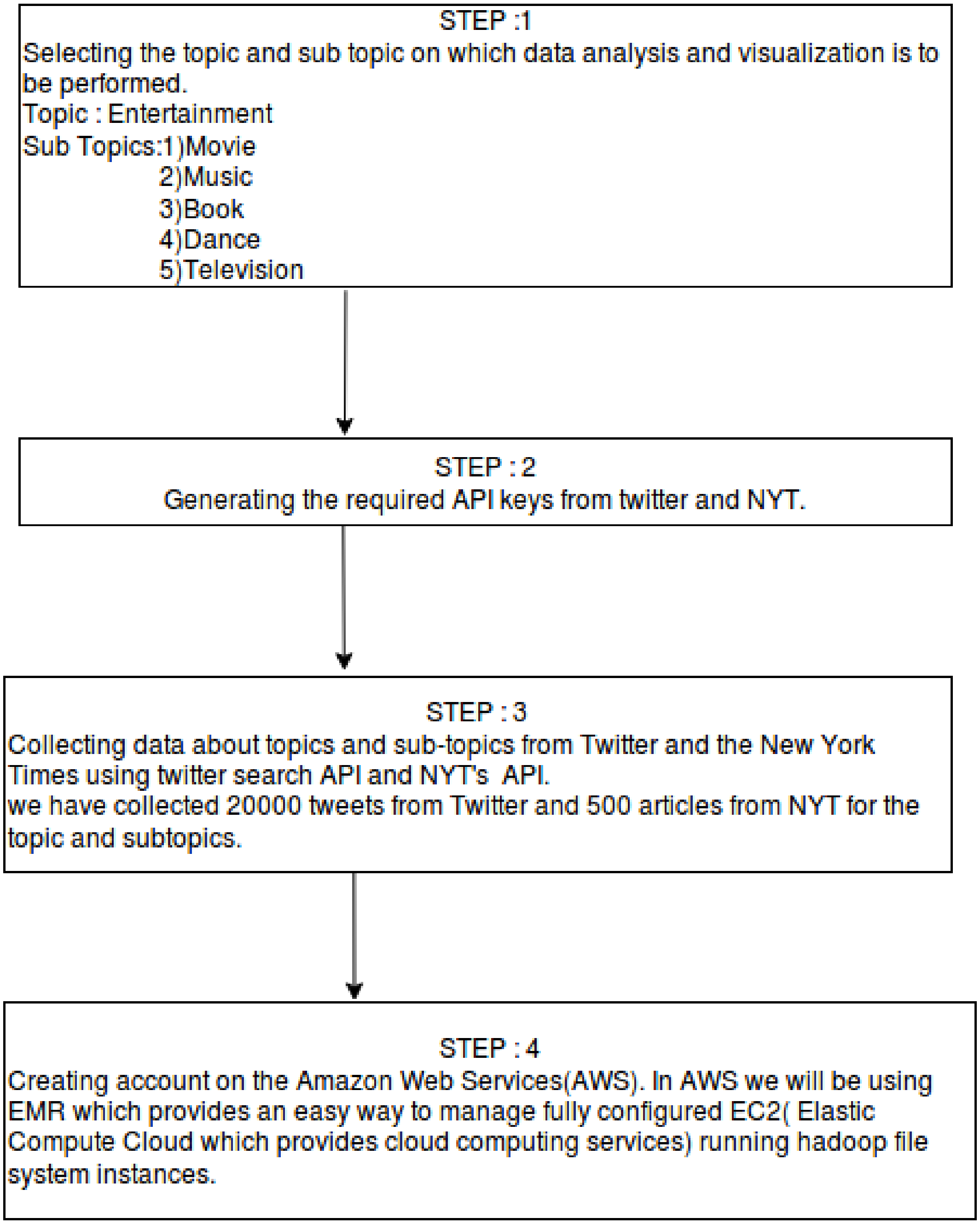
# DATA AGGREGATION,BIG DATA ANALYSIS AND VISUALIZATION

**TANMAY SINGH (UBIT : tanmaypr , Person No: 50291086)**

**PRANJAL JAIN (UBIT : pjain5, Person No: 50289299)**

**Website Link:--https://dic2.000webhostapp.com/DIC.html**

# Flow Diagram

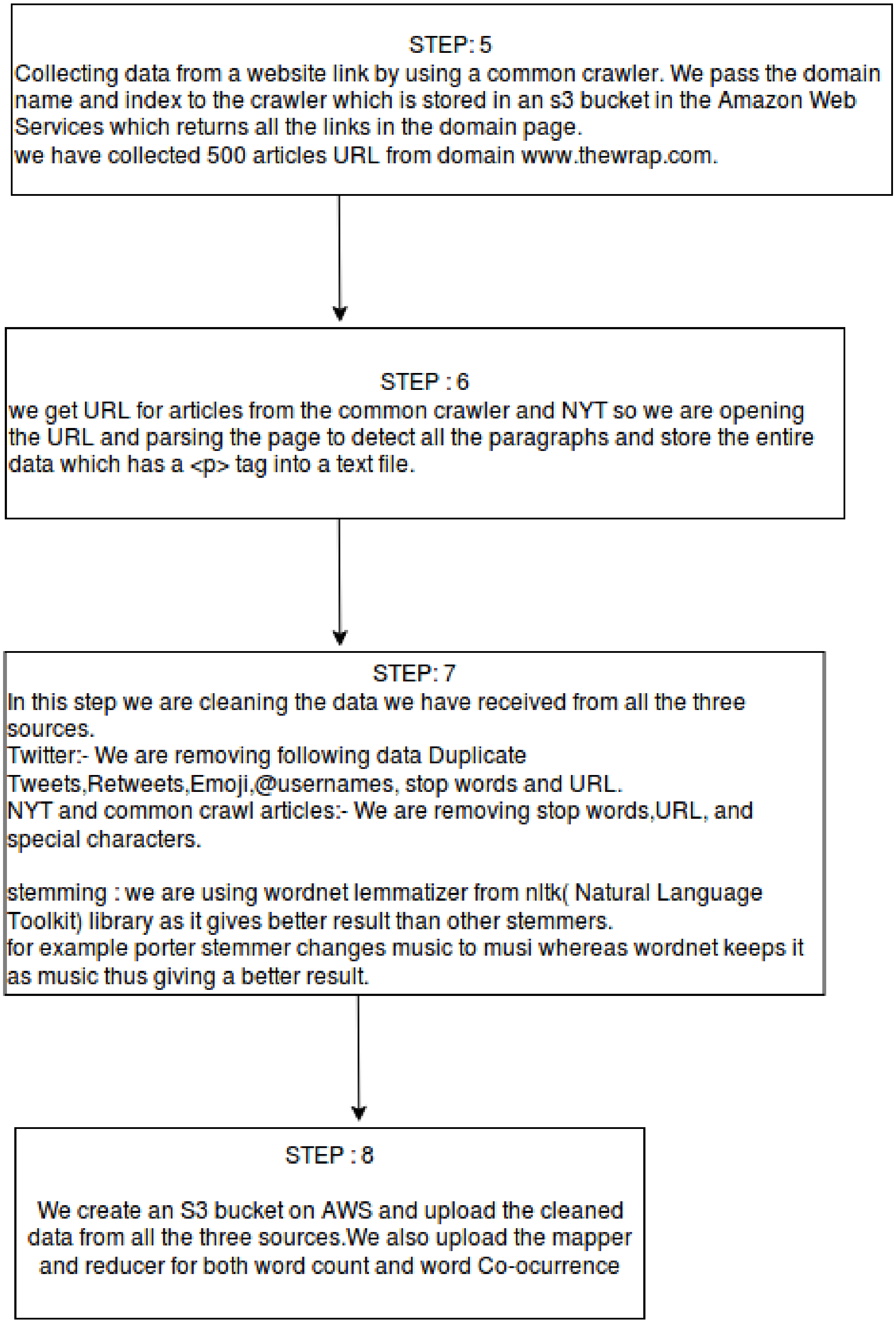


STEP :1
Selecting the topic and sub topic on which data analysis and visualization is to be performed.
Topic : Entertainment
Sub Topics:1)Movie
2)Music
3)Book
4)Dance
5)Television

STEP : 2
Generating the required API keys from twitter and NYT.

STEP : 3
Collecting data about topics and sub-topics from Twitter and the New York Times using twitter search API and NYT's API.
we have collected 20000 tweets from Twitter and 500 articles from NYT for the topic and subtopics.

STEP : 4
Creating account on the Amazon Web Services(AWS). In AWS we will be using EMR which provides an easy way to manage fully configured EC2( Elastic Compute Cloud which provides cloud computing services) running hadoop file system instances.



STEP: 5

Collecting data from a website link by using a common crawler. We pass the domain name and index to the crawler which is stored in an s3 bucket in the Amazon Web Services which returns all the links in the domain page.

we have collected 500 articles URL from domain www.thewrap.com.

STEP : 6

we get URL for articles from the common crawler and NYT so we are opening the URL and parsing the page to detect all the paragraphs and store the entire data which has a <p> tag into a text file.

STEP: 7

In this step we are cleaning the data we have received from all the three sources.

Twitter:- We are removing following data Duplicate Tweets,Retweets,Emoji,@usernames, stop words and URL.

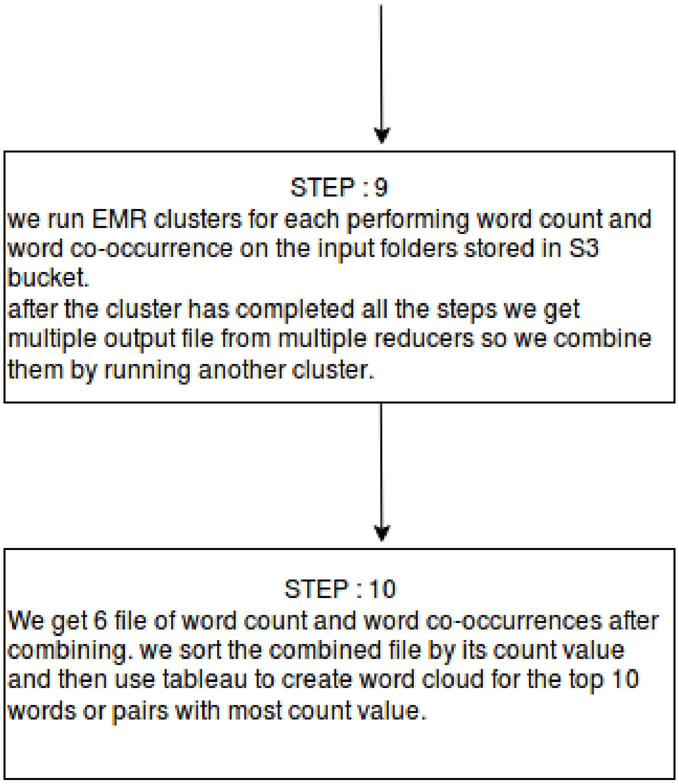
NYT and common crawl articles:- We are removing stop words,URL, and special characters.

stemming : we are using wordnet lemmatizer from nltk( Natural Language Toolkit) library as it gives better result than other stemmers.

for example porter stemmer changes music to musi whereas wordnet keeps it as music thus giving a better result.

STEP : 8

We create an S3 bucket on AWS and upload the cleaned data from all the three sources.We also upload the mapper and reducer for both word count and word Co-ocurrence



STEP : 9

we run EMR clusters for each performing word count and word co-occurrence on the input folders stored in S3 bucket.

after the cluster has completed all the steps we get multiple output file from multiple reducers so we combine them by running another cluster.

STEP : 10

We get 6 file of word count and word co-occurrences after combining. we sort the combined file by its count value and then use tableau to create word cloud for the top 10 words or pairs with most count value.

## Directory Structure S3 AWS

- S3
  - diclab2commoncrawl
    - crawler input
    - crawler word count
    - crawler word count combined
    - crawler co-occur
    - crawler co-occur combined
    - crawler-mapper.py
    - crawler-reducer.py
  - diclab2twitter
    - same structure as diclab2commoncrawl
  - diclab2nyt
    - same directory structure as diclab2commoncrawl

## Libraries Used

-Tweepy
-nltk
-urllib.request
-nytimesarticle
-requests
-json
-gzip
-zlib

## Reference

-https://www.bellingcat.com/resource/2015/08/13/using-python-to-mine-common-crawl