Natural Language Processing: Approaches to Text Summarization

Geoffrey Harper V00866723

Introduction

The following report provides a general overview of Text Summarization apporaches and techniques as well as provides as an implementation of the EdgeSumm text summarization system [5]

What is text summarization and it purpose

Summarizing text in NLP is the proccess of taking a given text an outputing a human-comprehensible summary of the given input. In the era of big data, where we have massive corpuses of data, producing succinct document summaries can be important for many areas of industry. For example, compling Amazon product reviews for a marketing team or having a research data-base present key-inforamtion of journal papers. However, since computers lack the trival understanding of human language creating text summareis that are coherent and correct becomes a very non-trival task [1].

Theory and techniques used in the NLP Text summarization field

Text summary can be viewed as a sequence mapping problem. It takes a sequences of words (the given text) and maps it to a new sequences of words (the summary text).

There are two broad main catagoreis of text summarization methods.

1) Abstractive summarization.

2) Extractive summarization.

Abstractive summarization approach creates a summary much like a human would make when abstracting a document. It uses the latest sequence mapping techniques in machine learning such as, encoding and decoding, as well as generative proccesses that will make human like sentances and rephrasings of a given article [1] [2].

Extractive summarization is a model in which the output sequence is a subset of the input text, containing what sentance of the article that are deemed to convey the input texts main ideas [1] [2].

A common approaches to text summariztion:

At the momemnt the cutting edge approach to text summariztion is using a very modern neural-net architecture type called Transformers [3]. This model produces very human like text summariztions.

However, as with the common issue with neural-nets, you needs massive amounts of data to train it successfully and even then there could be underlying hidden variables in the data set that can produce unpredicatble resutls in a generalized setting, with no way to look 'under-the-hood' of the neural-net and figure out what the precise problem is.

Another solution is the method known as Automatic-Text-Summarization (ATS). An ATS can be purly extractive, abstractive or a hybrid of each technique [4]. The basic architecture for ATS can be viewed in the figure below.

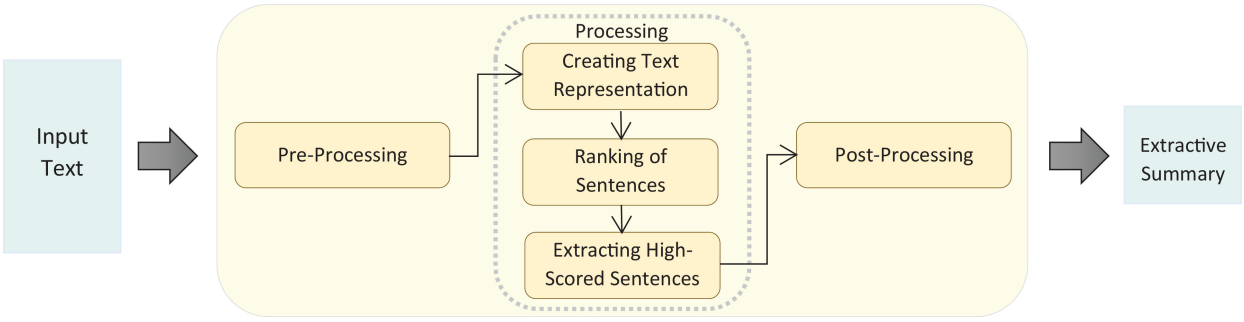

Figure 1: Basic architecture of an ATS [5].

EdgeSumm ATS

For this report the EdgeSumm ATS will be examined based on research in [5] and its subsequent code based provided [6].

How does EdgeSumm work?

EdgeSumm ATS was published in 2020 by Wafaa S.El-Kassas Et al. as a novel approach to text summarization. The system proposes a new method for extractive single document summariztion using a graph based approach. It attempts to combine a number to extractive methods to benifit from the advantages they provide, as well as propose a new graph model that updates node weights dynamically while the system is being run [5].

Wafaa's paper presents the EdgeSumm system implmentation as follows.

1) Input and pre-processing

```
    a) Input canidate length you want for your text summary and input
your text.

    b) Data is pre-processed
```

2) Process data

```
    c) Construction of a text graph model representation for the input
document then proceeds.
```

d) Two graph search algorhtims that search constructed graph sentences to be included in candidate summary.

e) Steps c)-d) interated over with the canidate summary acting as the new input text until a summary is created that is less-than-or-equal-to the input summary length.

f) If the canidate summary is not converging towards the specified input length then the interated until the generated summary length is stabalized.

3) Post Processing

a) when summary length has not been met implment a sentence selection algorthim to reduce summary length.

b) Reorder summary sentences based on input textorder.

c) Concatinate sentences and output the final text summary

Dataset

The datasets used in the paper for testing and validation where the DUC200 and DUC2001. However, this project will test the system on two very differing datasets. The first dataset is The New York Times Annotated Corpus (NYTAC), and the second dataset is the SAMSum dataset. The NYTAC is a dataset that although isn't written in pure logical preposition is still held to high grammtical english language sentence structure so I belive that the EdgeSumm will produce strong results. SAMSum, on the other hand, is quite the opposite use of the english language. SAMSum is a dataset of text message converstations. Not only does the conversations not flow smoothly, unlike and professional journalists article would, but, ranking the importantce of sentances might be difficult due to the conversation like flow of the messages.

Loading the Dataset

Read in the reddit Tifu json dataset

In [4]:
```
import json

posts = []
with open('tifu_all_tokenized_and_filtered.json', 'r') as fp:
    for line in fp:
        posts.append(json.loads(line))

# Json entries
print(posts[50000].keys())
```

```
dict_keys(['title_tokenized', 'permalink', 'title', 'url', 'num_comments', 'tldr', 'crea
ted_utc', 'trimmed_title_tokenized', 'id', 'selftext_html', 'score', 'upvote_ratio', 'tl
```

```
dr_tokenized', 'selftext', 'trimmed_title', 'selftext_without_tldr_tokenized', 'ups', 's
elftext_without_tldr'])
```

In [ ]:

In [5]:
```
!pip install beautifulsoup4 markdown
```

```
Requirement already satisfied: beautifulsoup4 in c:\users\inbox\anaconda3\lib\site-packa
ges (4.10.0)
Requirement already satisfied: markdown in c:\users\inbox\anaconda3\lib\site-packages
(3.3.6)
Requirement already satisfied: soupsieve>1.2 in c:\users\inbox\anaconda3\lib\site-packag
es (from beautifulsoup4) (2.2.1)
Requirement already satisfied: importlib-metadata>=4.4 in c:\users\inbox\anaconda3\lib\s
ite-packages (from markdown) (4.8.1)
Requirement already satisfied: zipp>=0.5 in c:\users\inbox\anaconda3\lib\site-packages
(from importlib-metadata>=4.4->markdown) (3.6.0)
```

Clean the dataset by removing all markdown

Not all data has a summary so only indluced data that has a summary attached for validation purposes.

In [6]:
```
from bs4 import BeautifulSoup
from markdown import markdown

cleaned_data = []

for p in posts:
    if p['tldr'] != None:
        html = markdown(p['selftext_without_tldr'])
        text = ''.join(BeautifulSoup(html).findAll(text=True))
        cleaned_data.append((text,p['title_tokenized'],p['tldr']))
```

Diagram of System Architecture

The figure below is a graphical representation of how the edgeSumm system works

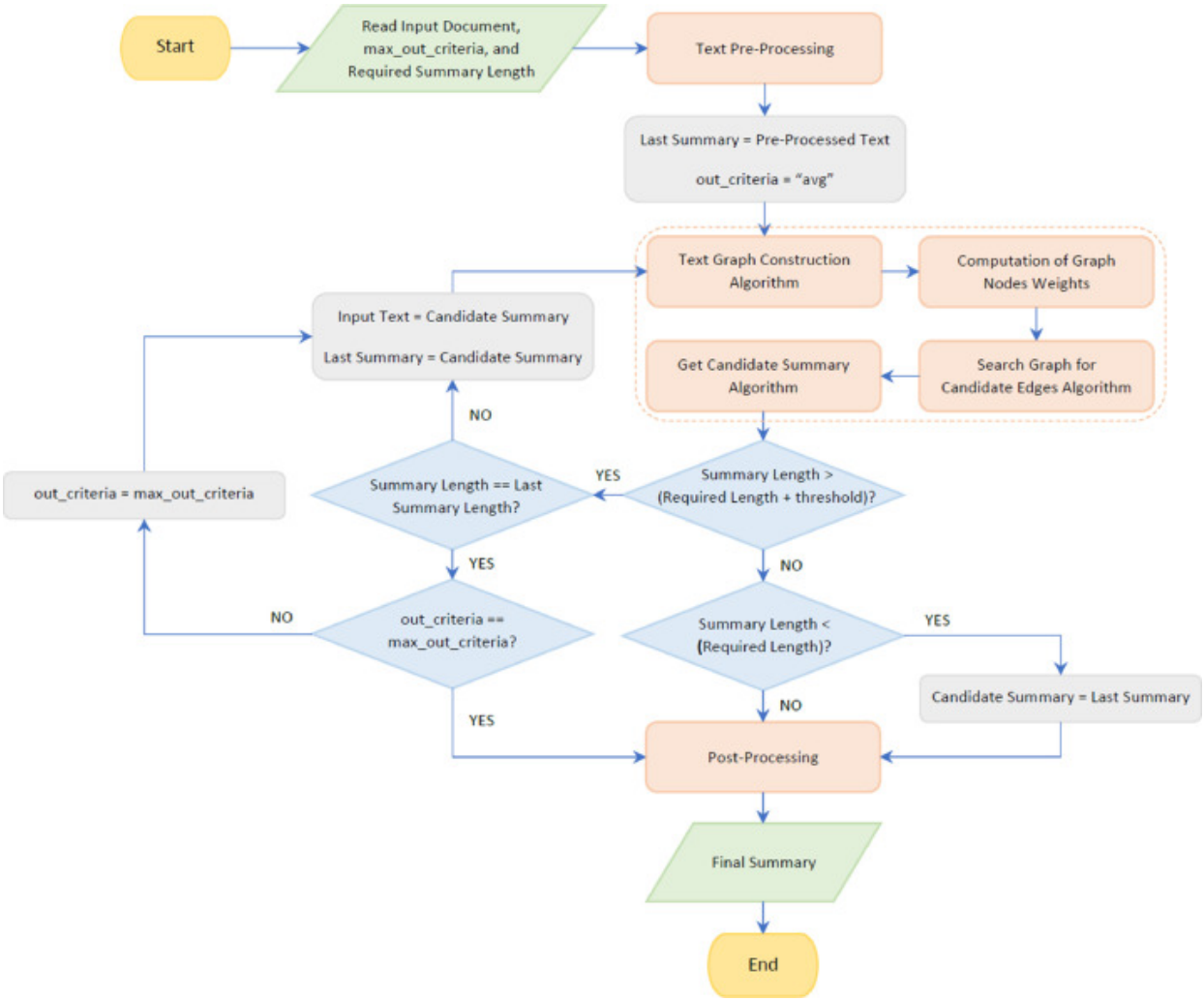

Figure 2: Basic architecture of an ATS [5].

Implmentation of System

The following is the implementation of the system

Pre-proccess the data

Data pre-processing. For data-pre-proccessing we will use a common library in the nlp world nltk and the python contractions library to deal with word contractions.

In [7]:
```
! pip install nltk contractions
```

```
Requirement already satisfied: nltk in c:\users\inbox\anaconda3\lib\site-packages (3.6.
5)
Requirement already satisfied: contractions in c:\users\inbox\anaconda3\lib\site-package
s (0.1.68)
Requirement already satisfied: click in c:\users\inbox\anaconda3\lib\site-packages (from
nltk) (8.0.3)
Requirement already satisfied: joblib in c:\users\inbox\anaconda3\lib\site-packages (fro
m nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\inbox\anaconda3\lib\site-pack
ages (from nltk) (2021.8.3)
```

```
Requirement already satisfied: tqdm in c:\users\inbox\anaconda3\lib\site-packages (from
nltk) (4.62.3)
Requirement already satisfied: textsearch>=0.0.21 in c:\users\inbox\anaconda3\lib\site-p
ackages (from contractions) (0.0.21)
Requirement already satisfied: pyahocorasick in c:\users\inbox\anaconda3\lib\site-packag
es (from textsearch>=0.0.21->contractions) (1.4.4)
Requirement already satisfied: anyascii in c:\users\inbox\anaconda3\lib\site-packages (f
rom textsearch>=0.0.21->contractions) (0.3.1)
Requirement already satisfied: colorama in c:\users\inbox\anaconda3\lib\site-packages (f
rom click->nltk) (0.4.4)
```

In [8]:

```
# pre-processing based on
# https://towardsdatascience.com/text-summarization-with-nlp-textrank-vs-seq2seq-vs-bar
#https://stackoverflow.com/questions/47274540/how-to-improve-nltk-sentence-segmentation
#https://www.nltk.org/_modules/nltk/tokenize/punkt.html

import re
import nltk
import contractions
from nltk.tokenize.punkt import PunktSentenceTokenizer, PunktParameters
from nltk.corpus import wordnet
from nltk.corpus import stopwords
import collections

import numpy as np


nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('punkt')

## create stopwords
lst_stopwords = stopwords.words("english")


### stemming (remove -ing, -ly, ...)
##ps = nltk.stem.porter.PorterStemmer()
##cleaned_sentences = [ps.stem(word) for word in cleaned_sentences]
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\inbox\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\inbox\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\inbox\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\inbox\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Text Manipulation according to what is decribed in the paper [5]

Sentence segmentation

In [9]:

```
#segment sentences using punkt nlp standard algorithm
```

```
def sentence_segmentation(text,text_title):
    tokenizer = PunktSentenceTokenizer()
    tokenizer.train(text)
    sentences = tokenizer.tokenize(text)
    title = text_title

    #remove hypens
    cleaned_sentences = [x.replace('-',' ') for x in sentences]
    #remove contractions
    cleaned_sentences = [contractions.fix(x) for x in cleaned_sentences ]
    cleaned_sentences = [re.sub(r'[^\w\s]', '', x) for x in cleaned_sentences]


    #tokenize each sentence
    #cleaned_sentences = [x.split() for x in cleaned_sentences]
    return cleaned_sentences, title


print(sentence_segmentation(cleaned_data[0][0],cleaned_data[0][1]))
print(cleaned_data[0][0])
```

```
(['this actually happened a couple of years ago', 'i grew up in germany where i went to
a german secondary school that went from 5th to 13th grade we still had 13 grades then t
hey have since changed that', 'my school was named after anne frank and we had a club th
at i was very active in from 9th grade on which was dedicated to teaching incoming 5th g
raders about anne franks life discrimination anti semitism hitler the third reich and th
at whole spiel', 'basically a day where the students classes are cancelled and instead w
e give them an interactive history and social studies class with lots of activities and
games', 'this was my last year at school and i already had a lot of experience doing the
se project days with the kids', 'i was running the thing with a friend so it was just th
e two of us and 30 something 5th graders', 'we start off with a brief introduction and b
rainstorming what do they know about anne frank and the third reich', 'you would be surp
rised how much they know', 'anyway after the brainstorming we do a few activities and th
en we take a short break', 'after the break we split the class into two groups to make i
t easier to handle', 'one group watches a short movie about anne frank while the other g
ets a tour through our poster presentation that our student group has been perfecting ov
er the years', 'then the groups switch', 'i am in the classroom to show my group the mov
ie and i take attendance to make sure no one decided to run away during break', 'i am go
ing down the list when i come to the name sandra name changed', 'a kid with a boyish hai
rcut and a somewhat deeper voice wearing clothes from the boys section at a big clothing
chain in germany pipes up', 'now keep in mind these are all 11 year olds they are all pr
e pubescent their bodies are not yet showing any sex specific features one would be able
to see while they are fully clothed eg', 'boobs beards', 'this being a 5th grade in the
rather conservative for german standards bavaria i was confused', 'i looked down at the
list again making sure i had read the name right', 'look back up at the kid', 'me you ar
e sandra', 'kid yep', 'me oh sorry', 'thinking the kid must be from somewhere where sand
ra is both a girls and boys name where are you from', 'i have only ever heard that as a
girls name before', 'the class starts laughing', 'sandra gets really quiet', 'i am a gir
l she says', 'some of the other students start saying that their parents made the same m
istake when they met sandra', 'i feel so sorry and stupid', 'i get the class to calm dow
n and finish taking attendance', 'we watch the movie in silence', 'after the movie when
we walked down to where the poster presentation took place i apologised to sandra', 'i f
elt so incredibly terrible i still do to this day', 'throughout the rest of the day i he
ard lots of whispers about sandra', 'i tried to stop them whenever they came up but ther
e was no stopping the 5th grade gossip i had set in motion', 'sandra if you are out ther
e i am so incredibly sorry for humiliating you in front of your class', 'i hope you are
happy and healthy and continue to live your life the way you like', 'do not let anyone t
ell you you have to dress or act a certain way just because of the body parts you were b
orn with', 'i am sorry if i made you feel like you were wrong for dressing and acting di
fferently', 'i am sorry i probably made that day hell for you', 'i am sorry for my ignor
```

```
ance'], ['tifu', 'by', 'gender', 'stereotyping'])
this actually happened a couple of years ago. i grew up in germany where i went to a ger
man secondary school that went from 5th to 13th grade (we still had 13 grades then, they
have since changed that). my school was named after anne frank and we had a club that i
was very active in from 9th grade on, which was dedicated to teaching incoming 5th grade
rs about anne franks life, discrimination, anti-semitism, hitler, the third reich and th
at whole spiel. basically a day where the students' classes are cancelled and instead we
give them an interactive history and social studies class with lots of activities and ga
mes.
this was my last year at school and i already had a lot of experience doing these projec
t days with the kids. i was running the thing with a friend, so it was just the two of u
s and 30-something 5th graders. we start off with a brief introduction and brainstormin
g: what do they know about anne frank and the third reich? you'd be surprised how much t
hey know. anyway after the brainstorming we do a few activities, and then we take a shor
t break. after the break we split the class into two groups to make it easier to handle.
one group watches a short movie about anne frank while the other gets a tour through our
poster presentation that our student group has been perfecting over the years. then the
groups switch.
i'm in the classroom to show my group the movie and i take attendance to make sure no on
e decided to run away during break. i'm going down the list when i come to the name sand
ra (name changed). a kid with a boyish haircut and a somewhat deeper voice, wearing clot
hes from the boy's section at a big clothing chain in germany, pipes up.
now keep in mind, these are all 11 year olds, they are all pre-pubescent, their bodies a
re not yet showing any sex specific features one would be able to see while they are ful
ly clothed (e.g. boobs, beards,...). this being a 5th grade in the rather conservative
(for german standards) bavaria, i was confused. i looked down at the list again making s
ure i had read the name right. look back up at the kid.
me: "you're sandra?"
kid: "yep."
me: "oh, sorry. thinking the kid must be from somewhere where sandra is both a girl's an
d boy's name where are you from? i've only ever heard that as a girl's name before."
the class starts laughing. sandra gets really quiet. "i am a girl..." she says. some of
the other students start saying that their parents made the same mistake when they met s
andra. i feel so sorry and stupid. i get the class to calm down and finish taking attend
ance. we watch the movie in silence. after the movie, when we walked down to where the p
oster presentation took place i apologised to sandra. i felt so incredibly terrible, i s
till do to this day. throughout the rest of the day i heard lots of whispers about sandr
a. i tried to stop them whenever they came up, but there was no stopping the 5th grade g
ossip i had set in motion.
sandra, if you're out there, i am so incredibly sorry for humiliating you in front of yo
ur class. i hope you are happy and healthy and continue to live your life the way you li
ke. don't let anyone tell you you have to dress or act a certain way just because of the
body parts you were born with. i'm sorry if i made you feel like you were wrong for dres
sing and acting differently. i'm sorry i probably made that day hell for you. i'm sorry
for my ignorance.
```

Remove stop-words

In [7]:

```
#remove stop-words (Words that can be removed and still keep the importance of the sent
def remove_stop_words (cleaned_sentences,title,lst_stopwords):
    txt = []
    for lst_txt in cleaned_sentences:
        lst_txt = [word for word in lst_txt if word not in lst_stopwords]
        txt.append(lst_txt)

    cleaned_sentences = txt
    title = [word for word in title if word not in lst_stopwords]

    cleaned_sentences.insert(0,title) #define title at the first sentence in the given
```

```
    return cleaned_sentences

cleaned_sentences, title = sentence_segmentation(cleaned_data[0][0],cleaned_data[0][1])

cleaned_sentences = remove_stop_words(cleaned_sentences,title,lst_stopwords)
```

Standardize synonyms

In [16]:

```
# standardize synonyms
# the edgeSumm paper describes this as being an important point for weighting tranisiti
# if there are synonyms in the texts words
# we want these synonyms to be the same so the edge weight calculations in the graph ar

# standarizing synonyms will be implemented by the following

# prepare list of synonyms for each word
# create a key-value list where the key is the word and the value is it's list of synon

def standardize_synonyms(cleaned_sentences):

    syns_list = []

    #create list of words
    word_lst = [inner for outer in cleaned_sentences for inner in outer]
    word_lst = list(set(word_lst)) #remove_duplicates

    # create dictionary of synonyms
    for word in word_lst:
        synonyms = []
        for syn in wordnet.synsets(word):
            for l in syn.lemmas():
                synonyms.append(l.name())
        syns_list.append(synonyms)

    synonyms_dict = dict(zip(word_lst,syns_list))

    normalized_words = []

    for lst_txt in cleaned_sentences: #pick the normailzed word to be the first synonym
        txt = []
        for i in range(len(lst_txt)):
            if len(synonyms_dict[lst_txt[i]]) != 0:
                lst_txt[i] = synonyms_dict[lst_txt[i]][0]
            txt.append(lst_txt[i])

        normalized_words.append(txt)

    return normalized_words

normalized_words = standardize_synonyms(cleaned_sentences)
```

In [17]:

```
#create bi-grams. bi-grams that are repeated more than once are added to a list of freq

def bi_grams_by_frequency(normalized_words):

    bi_grams = list(nltk.bigrams(normalized_words))
```

```
    flattened_bi_grams = [item for sublist in bi_grams for item in sublist]

    tupled_bi_grams = [tuple(x) for x in flattened_bi_grams]


    #build frequency dictionary
    freq = {}

    for item in tupled_bi_grams:
        if (item in freq):
            freq[item] += 1
        else:
            freq[item] = 1

    #filter for bi-grams that appear more than once
    bi_grams_freq = [ (list(tup),freq[tup]) for tup in freq if freq[tup] > 1]

    return bi_grams_freq, bi_grams

bi_grams_freq, bi_grams = bi_grams_by_frequency(normalized_words)
```

In [18]:

```
#tag bi-grams by word type and filter so only adjectives and nouns stay
#this is used for graph node-wieghting

#input a bi_gram list where each tuple is are a list of bi-grams
def tag_bi_grams(bi_grams):

    tagged_sentences = [nltk.pos_tag(sentence[0]) for sentence in bi_grams]

    #print(tagged_sentences)

    bi_gram_tags = [item for sublist in tagged_sentences for item in sublist  if item[1

    bi_gram_tags = []


    for bi_gram in tagged_sentences:
        lst = []
        for w in bi_gram:
            if w[1] == 'JJ' or w[1] == 'NN':
                lst.append(w)
        bi_gram_tags.append(lst)

    return bi_gram_tags

bi_gram_tags = tag_bi_grams(bi_grams)
```

Word frequency of the text.

In the paper the word frequency is determined by the following.

Word freq = word_freq_given_document_title + word_freq_given_keywords + word_freq_given_list_of_sentences

In [19]:

```
def word_frequency(normalized_words,keywords):
    words_in_title = normalized_words[0]
```

```
    words_in_text = [item for sublist in range(1,len(normalized_words)) for item in nor

    ### Lemmatization (convert the word into root word)
    lem = nltk.stem.wordnet.WordNetLemmatizer()
    words_in_text = [lem.lemmatize(word) for word in words_in_text]
    words_in_title = [lem.lemmatize(word) for word in words_in_title]


    frequency = {}

    # iterating over the list
    for item in words_in_text:
       # checking the element in dictionary
        if item in frequency:
        # incrementing the count
            frequency[item] += 1
        else:
            # initializing the count
            frequency[item] = 1
        if item in words_in_title:
            frequency[item] += 1
        if item in keywords:
            frequency[item] += 1

    return frequency
```

Putting it all together.

Below we preform all the text-manipulation steps needed to run the EdgeSumm algorithm. What will be returned will be a list of nouns and adjectives for each bi-gram, bi-grams with frequency greater than 1, a dictionary of sentences in zero-based numbering format with the 0th sentence being the title of the text, if the title exists.

In [12]:

```
def text_manipulation(text,text_title,keywords):
    cleaned_sentences, title = sentence_segmentation(text,text_title)

    segmented_sentences = cleaned_sentences

    #tokenize each sentence
    cleaned_sentences = [x.split() for x in cleaned_sentences]
    cleaned_sentences = remove_stop_words(cleaned_sentences,title,lst_stopwords)

    normalized_words = standardize_synonyms(cleaned_sentences)

    bi_grams_freq, bi_grams = bi_grams_by_frequency(normalized_words)

    bi_gram_tags = tag_bi_grams(bi_grams)

    word_freq = word_frequency(normalized_words,keywords)


    #create sentences dictionary
    lst = range(len(normalized_words))
    sentences_dict = dict(zip(lst, normalized_words))

    lst = range(len(segmented_sentences))
```

```
    segmented_sentences = dict(zip(lst,segmented_sentences))

    return sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences

sentences_dict, word_freq, bi_grams_freq, bi_gram_tags,segmented_sentences = text_manip
```

Graph construction and graph iteration proccess

After text has been pre-processed a graphical representation is created from the input text with edge weights.

The following code implements based on [5] and [6].

Build graph

The following implments the pseudo-code in the figure below

Algorithm 1 Text Graph Construction Algorithm

```
Input: t, K, S, F, BI, P, B
Output: G
Variables: source_node = "S#" , edge_label = "" , edge_order = 0, destination_node = ""
// Each entry in the S list is a key-value pair: the key is the sentence code and the value is the sentence text.
for each s in S
    // s.key stores the sentence unique code and s.value stores the sentence text.
    words = tokenization(s.value)
    tags = POS(words)
    for each word in words
        if tags[word] is noun then
            word_lemma= lemmatization(word)
            destination_node = word_lemma
            add_node(word_lemma, s.key, word)
            add_edge(source_node, destination_node, edge_label, s.key, edge_order)
            edge_order = edge_order + 1
            source_node = word_lemma
            destination_node = ""
            edge_label = ""
        else if tags[word] is not noun then
            edge_label = edge_label + " " + word
        end
    end
    add_edge(source_node, "E#", edge_label, s.key, edge_order)
    edge_order = 0
    source_node = "S#"
    edge_label = ""
    destination_node = ""
end
```

Figure 3 graph construction [5]

input:

- sentence dictionary S
- title of document t
- computed word-frequency F

```
- frequent bi-grams BI
- nouns of document P
- list of keywords K
- list of domain specific or biased words B
```

output a text graph G

note by definition of how the data was processed in the first sentence is the title

In [13]:
```
!pip install networkx
#use this library for graph building
```

```
Requirement already satisfied: networkx in c:\users\inbox\anaconda3\lib\site-packages
(2.6.3)
```

In [14]:
```
import networkx as nx

def text_graph_construction(S):
    G = nx.Graph()
    t = S[0]
    S.pop(0)
    for key in S:
        edge_order = 0
        source_node = 'S#'
        edge_label = ''
        destination_node = ''
        words = S[key]
        tagged_words = nltk.pos_tag(words)
        for word in tagged_words:
            if word[1] == 'NN': #word is a noun
                lem = nltk.stem.wordnet.WordNetLemmatizer()
                lem_word = lem.lemmatize(word[0])
                destination_node = lem_word
                G.add_node(lem_word,key = key,word= word[0], weight = 0)
                G.add_edge(source_node,destination_node,edge_label=edge_label,key=key,e
                edge_order += 1
                source_node = lem_word
                edge_label = ''
                destination_node = ''
            elif word[1] == 'JJ': #word is adjective
                edge_label = word[0]
        G.add_edge(source_node,"#E",edge_label=edge_label,key=key,edge_order=edge_order


    return G



keywords = []
domain_words = []
sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = text_mani
G = text_graph_construction(sentences_dict)

print(list(G.nodes(data=True)))
print(list(G.edges(data=True)))
```

[('couple', {'key': 1, 'word': 'couple', 'weight': 0}), ('S#', {}), ('old_age', {'key': 11, 'word': 'old_age', 'weight': 0}), ('#E', {}), ('turn', {'key': 2, 'word': 'turn', 'weight': 0}), ('school', {'key': 5, 'word': 'school', 'weight': 0}), ('travel', {'key': 2, 'word': 'travel', 'weight': 0}), ('class', {'key': 37, 'word': 'class', 'weight': 0}), ('change', {'key': 14, 'word': 'change', 'weight': 0}), ('name', {'key': 25, 'word': 'name', 'weight': 0}), ('entrance', {'key': 3, 'word': 'entrance', 'weight': 0}), ('grader', {'key': 6, 'word': 'grader', 'weight': 0}), ('life', {'key': 38, 'word': 'life', 'weight': 0}), ('discrimination', {'key': 3, 'word': 'discrimination', 'weight': 0}), ('semitism', {'key': 3, 'word': 'semitism', 'weight': 0}), ('one-third', {'key': 3, 'word': 'one-third', 'weight': 0}), ('spiel', {'key': 3, 'word': 'spiel', 'weight': 0}), ('day', {'key': 41, 'word': 'day', 'weight': 0}), ('student', {'key': 29, 'word': 'student', 'weight': 0}), ('cancel', {'key': 4, 'word': 'cancel', 'weight': 0}), ('history', {'key': 4, 'word': 'history', 'weight': 0}), ('survey', {'key': 4, 'word': 'survey', 'weight': 0}), ('activity', {'key': 9, 'word': 'activity', 'weight': 0}), ('game', {'key': 4, 'word': 'game', 'weight': 0}), ('stopping_point', {'key': 5, 'word': 'stopping_point', 'weight': 0}), ('year', {'key': 16, 'word': 'year', 'weight': 0}), ('experience', {'key': 5, 'word': 'experience', 'weight': 0}), ('child', {'key': 24, 'word': 'child', 'weight': 0}), ('thing', {'key': 6, 'word': 'thing', 'weight': 0}), ('friend', {'key': 6, 'word': 'friend', 'weight': 0}), ('something', {'key': 6, 'word': 'something', 'weight': 0}), ('start', {'key': 29, 'word': 'start', 'weight': 0}), ('brief', {'key': 7, 'word': 'brief', 'weight': 0}), ('introduction', {'key': 7, 'word': 'introduction', 'weight': 0}), ('interruption', {'key': 13, 'word': 'interruption', 'weight': 0}), ('split', {'key': 10, 'word': 'split', 'weight': 0}), ('group', {'key': 13, 'word': 'group', 'weight': 0}), ('brand', {'key': 13, 'word': 'brand', 'weight': 0}), ('movie', {'key': 33, 'word': 'movie', 'weight': 0}), ('poster', {'key': 33, 'word': 'poster', 'weight': 0}), ('presentation', {'key': 33, 'word': 'presentation', 'weight': 0}), ('switch', {'key': 12, 'word': 'switch', 'weight': 0}), ('classroom', {'key': 13, 'word': 'classroom', 'weight': 0}), ('show', {'key': 13, 'word': 'show', 'weight': 0}), ('return', {'key': 13, 'word': 'return', 'weight': 0}), ('attendance', {'key': 31, 'word': 'attendance', 'weight': 0}), ('decide', {'key': 13, 'word': 'decide', 'weight': 0}), ('departure', {'key': 14, 'word': 'departure', 'weight': 0}), ('list', {'key': 19, 'word': 'list', 'weight': 0}), ('haircut', {'key': 15, 'word': 'haircut', 'weight': 0}), ('voice', {'key': 15, 'word': 'voice', 'weight': 0}), ('erosion', {'key': 15, 'word': 'erosion', 'weight': 0}), ('apparel', {'key': 15, 'word': 'apparel', 'weight': 0}), ('male_child', {'key': 24, 'word': 'male_child', 'weight': 0}), ('section', {'key': 15, 'word': 'section', 'weight': 0}), ('clothing', {'key': 15, 'word': 'clothing', 'weight': 0}), ('chain', {'key': 15, 'word': 'chain', 'weight': 0}), ('pipe', {'key': 15, 'word': 'pipe', 'weight': 0}), ('support', {'key': 16, 'word': 'support', 'weight': 0}), ('mind', {'key': 16, 'word': 'mind', 'weight': 0}), ('pre', {'key': 16, 'word': 'pre', 'weight': 0}), ('pubescent', {'key': 16, 'word': 'pubescent', 'weight': 0}), ('body', {'key': 39, 'word': 'body', 'weight': 0}), ('sexual_activity', {'key': 16, 'word': 'sexual_activity', 'weight': 0}), ('feature', {'key': 16, 'word': 'feature', 'weight': 0}), ('eg', {'key': 16, 'word': 'eg', 'weight': 0}), ('standard', {'key': 18, 'word': 'standard', 'weight': 0}), ('confuse', {'key': 18, 'word': 'confuse', 'weight': 0}), ('look', {'key': 19, 'word': 'look', 'weight': 0}), ('expression', {'key': 20, 'word': 'expression', 'weight': 0}), ('sandra', {'key': 37, 'word': 'sandra', 'weight': 0}), ('yep', {'key': 22, 'word': 'yep', 'weight': 0}), ('regretful', {'key': 41, 'word': 'regretful', 'weight': 0}), ('girl', {'key': 25, 'word': 'girl', 'weight': 0}), ('parent', {'key': 29, 'word': 'parent', 'weight': 0}), ('mistake', {'key': 29, 'word': 'mistake', 'weight': 0}), ('meet', {'key': 29, 'word': 'meet', 'weight': 0}), ('composure', {'key': 31, 'word': 'composure', 'weight': 0}), ('watch', {'key': 32, 'word': 'watch', 'weight': 0}), ('silence', {'key': 32, 'word': 'silence', 'weight': 0}), ('topographic_point', {'key': 33, 'word': 'topographic_point', 'weight': 0}), ('felt', {'key': 34, 'word': 'felt', 'weight': 0}), ('fillet', {'key': 36, 'word': 'fillet', 'weight': 0}), ('chitchat', {'key': 36, 'word': 'chitchat', 'weight': 0}), ('gesture', {'key': 36, 'word': 'gesture', 'weight': 0}), ('humiliate', {'key': 37, 'word': 'humiliate', 'weight': 0}), ('front', {'key': 37, 'word': 'front', 'weight': 0}), ('hope', {'key': 38, 'word': 'hope', 'weight': 0}), ('continue', {'key': 38, 'word': 'continue', 'weight': 0}), ('manner', {'key': 39, 'word': 'manner', 'weight': 0}), ('anyone', {'key': 39, 'word': 'anyone', 'weight': 0}), ('dress', {'key': 39, 'word': 'dress', 'weight': 0}), ('act', {'key': 39, 'word': 'act', 'weight': 0}), ('feel', {'key': 40, 'w

ord': 'feel', 'weight': 0}), ('ignorance', {'key': 42, 'word': 'ignorance', 'weight': 0})]
[('couple', 'S#', {'edge_label': 'happen', 'key': 1, 'edge_order': 0}), ('couple', 'old_age', {'edge_label': '', 'key': 1, 'edge_order': 1}), ('S#', 'turn', {'edge_label': '', 'key': 2, 'edge_order': 0}), ('S#', 'school', {'edge_label': '', 'key': 3, 'edge_order': 0}), ('S#', 'day', {'edge_label': '', 'key': 35, 'edge_order': 0}), ('S#', 'stopping_point', {'edge_label': '', 'key': 5, 'edge_order': 0}), ('S#', 'thing', {'edge_label': '', 'key': 6, 'edge_order': 0}), ('S#', 'start', {'edge_label': '', 'key': 7, 'edge_order': 0}), ('S#', '#E', {'edge_label': 'stupid', 'key': 30, 'edge_order': 0}), ('S#', 'activity', {'edge_label': 'brainstorming', 'key': 9, 'edge_order': 0}), ('S#', 'interruption', {'edge_label': '', 'key': 10, 'edge_order': 0}), ('S#', 'group', {'edge_label': '', 'key': 12, 'edge_order': 0}), ('S#', 'classroom', {'edge_label': '', 'key': 13, 'edge_order': 0}), ('S#', 'departure', {'edge_label': '', 'key': 14, 'edge_order': 0}), ('S#', 'child', {'edge_label': '', 'key': 24, 'edge_order': 0}), ('S#', 'support', {'edge_label': '', 'key': 16, 'edge_order': 0}), ('S#', 'class', {'edge_label': '', 'key': 31, 'edge_order': 0}), ('S#', 'look', {'edge_label': '', 'key': 19, 'edge_order': 0}), ('S#', 'expression', {'edge_label': '', 'key': 20, 'edge_order': 0}), ('S#', 'sandra', {'edge_label': '', 'key': 37, 'edge_order': 0}), ('S#', 'regretful', {'edge_label': '', 'key': 41, 'edge_order': 0}), ('S#', 'girl', {'edge_label': 'hear', 'key': 25, 'edge_order': 0}), ('S#', 'student', {'edge_label': '', 'key': 29, 'edge_order': 0}), ('S#', 'watch', {'edge_label': '', 'key': 32, 'edge_order': 0}), ('S#', 'movie', {'edge_label': '', 'key': 33, 'edge_order': 0}), ('S#', 'felt', {'edge_label': '', 'key': 34, 'edge_order': 0}), ('S#', 'fillet', {'edge_label': 'come', 'key': 36, 'edge_order': 0}), ('S#', 'hope', {'edge_label': '', 'key': 38, 'edge_order': 0}), ('S#', 'anyone', {'edge_label': '', 'key': 39, 'edge_order': 0}), ('S#', 'feel', {'edge_label': 'regretful', 'key': 40, 'edge_order': 0}), ('S#', 'ignorance', {'edge_label': 'regretful', 'key': 42, 'edge_order': 0}), ('old_age', '#E', {'edge_label': '', 'key': 11, 'edge_order': 7}), ('old_age', 'group', {'edge_label': 'perfect', 'key': 11, 'edge_order': 6}), ('#E', 'change', {'edge_label': '', 'key': 14, 'edge_order': 4}), ('#E', 'spiel', {'edge_label': '', 'key': 3, 'edge_order': 10}), ('#E', 'game', {'edge_label': '', 'key': 4, 'edge_order': 9}), ('#E', 'child', {'edge_label': '', 'key': 20, 'edge_order': 2}), ('#E', 'grader', {'edge_label': '', 'key': 6, 'edge_order': 4}), ('#E', 'introduction', {'edge_label': 'one-third', 'key': 7, 'edge_order': 3}), ('#E', 'interruption', {'edge_label': '', 'key': 13, 'edge_order': 9}), ('#E', 'brand', {'edge_label': 'handle', 'key': 10, 'edge_order': 5}), ('#E', 'switch', {'edge_label': '', 'key': 12, 'edge_order': 2}), ('#E', 'pipe', {'edge_label': '', 'key': 15, 'edge_order': 10}), ('#E', 'eg', {'edge_label': '', 'key': 16, 'edge_order': 9}), ('#E', 'confuse', {'edge_label': '', 'key': 18, 'edge_order': 3}), ('#E', 'name', {'edge_label': '', 'key': 25, 'edge_order': 2}), ('#E', 'sandra', {'edge_label': '', 'key': 35, 'edge_order': 2}), ('#E', 'yep', {'edge_label': '', 'key': 22, 'edge_order': 2}), ('#E', 'regretful', {'edge_label': '', 'key': 23, 'edge_order': 1}), ('#E', 'start', {'edge_label': '', 'key': 26, 'edge_order': 2}), ('#E', 'attendance', {'edge_label': '', 'key': 31, 'edge_order': 3}), ('#E', 'silence', {'edge_label': '', 'key': 32, 'edge_order': 3}), ('#E', 'topographic_point', {'edge_label': '', 'key': 33, 'edge_order': 4}), ('#E', 'day', {'edge_label': '', 'key': 41, 'edge_order': 2}), ('#E', 'gesture', {'edge_label': '', 'key': 36, 'edge_order': 4}), ('#E', 'class', {'edge_label': '', 'key': 37, 'edge_order': 4}), ('#E', 'manner', {'edge_label': '', 'key': 38, 'edge_order': 4}), ('#E', 'body', {'edge_label': '', 'key': 39, 'edge_order': 5}), ('#E', 'feel', {'edge_label': 'wrong', 'key': 40, 'edge_order': 1}), ('#E', 'ignorance', {'edge_label': '', 'key': 42, 'edge_order': 1}), ('turn', 'school', {'edge_label': 'secondary', 'key': 2, 'edge_order': 1}), ('school', 'travel', {'edge_label': '', 'key': 2, 'edge_order': 2}), ('school', 'name', {'edge_label': '', 'key': 3, 'edge_order': 1}), ('school', 'year', {'edge_label': '', 'key': 5, 'edge_order': 2}), ('school', 'experience', {'edge_label': '', 'key': 5, 'edge_order': 3}), ('travel', 'class', {'edge_label': 'thirteenth', 'key': 2, 'edge_order': 3}), ('class', 'class', {'edge_label': 'thirteen', 'key': 2, 'edge_order': 4}), ('class', 'change', {'edge_label': '', 'key': 2, 'edge_order': 5}), ('class', 'name', {'edge_label': 'ninth', 'key': 3, 'edge_order': 2}), ('class', 'entrance', {'edge_label': '', 'key': 3, 'edge_order': 3}), ('class', 'student', {'edge_label': '', 'key': 4, 'edge_order': 2}), ('class', 'cancel', {'edge_label': '', 'key': 4, 'edge_order': 3}), ('class', 'survey', {'edge_label': '', 'key': 4, 'edge_order': 6}), ('class', 'activity', {'edge_label': '', 'key': 4, 'edge_order': 7}), ('class', 'split', {'edge_label':

'', 'key': 10, 'edge_order': 2}), ('class', 'group', {'edge_label': '', 'key': 10, 'edge
_order': 3}), ('class', 'standard', {'edge_label': 'German', 'key': 18, 'edge_order':
1}), ('class', 'start', {'edge_label': '', 'key': 26, 'edge_order': 1}), ('class', 'comp
osure', {'edge_label': '', 'key': 31, 'edge_order': 1}), ('class', 'fillet', {'edge_labe
l': 'fifth', 'key': 36, 'edge_order': 1}), ('class', 'chitchat', {'edge_label': '', 'ke
y': 36, 'edge_order': 2}), ('class', 'front', {'edge_label': '', 'key': 37, 'edge_orde
r': 3}), ('change', 'name', {'edge_label': '', 'key': 14, 'edge_order': 3}), ('name', 'l
ist', {'edge_label': 'read', 'key': 19, 'edge_order': 2}), ('name', 'male_child', {'edge
_label': '', 'key': 24, 'edge_order': 3}), ('name', 'girl', {'edge_label': '', 'key': 2
5, 'edge_order': 1}), ('entrance', 'grader', {'edge_label': 'fifth', 'key': 3, 'edge_ord
er': 4}), ('grader', 'life', {'edge_label': '', 'key': 3, 'edge_order': 5}), ('grader',
'something', {'edge_label': 'fifth', 'key': 6, 'edge_order': 3}), ('life', 'discriminati
on', {'edge_label': '', 'key': 3, 'edge_order': 6}), ('life', 'continue', {'edge_label':
'populate', 'key': 38, 'edge_order': 2}), ('life', 'manner', {'edge_label': '', 'key': 3
8, 'edge_order': 3}), ('discrimination', 'semitism', {'edge_label': '', 'key': 3, 'edge_
order': 7}), ('semitism', 'one-third', {'edge_label': '', 'key': 3, 'edge_order': 8}),
('one-third', 'spiel', {'edge_label': '', 'key': 3, 'edge_order': 9}), ('day', 'studen
t', {'edge_label': '', 'key': 4, 'edge_order': 1}), ('day', 'felt', {'edge_label': 'awfu
l', 'key': 34, 'edge_order': 1}), ('day', 'sandra', {'edge_label': '', 'key': 35, 'edge_
order': 1}), ('day', 'regretful', {'edge_label': '', 'key': 41, 'edge_order': 1}), ('stu
dent', 'presentation', {'edge_label': '', 'key': 11, 'edge_order': 4}), ('student', 'gro
up', {'edge_label': '', 'key': 11, 'edge_order': 5}), ('student', 'start', {'edge_labe
l': '', 'key': 29, 'edge_order': 1}), ('cancel', 'history', {'edge_label': 'synergisti
c', 'key': 4, 'edge_order': 4}), ('history', 'survey', {'edge_label': 'sociable', 'key':
4, 'edge_order': 5}), ('activity', 'game', {'edge_label': '', 'key': 4, 'edge_order':
8}), ('activity', 'interruption', {'edge_label': 'short', 'key': 9, 'edge_order': 1}),
('stopping_point', 'year', {'edge_label': '', 'key': 5, 'edge_order': 1}), ('year', 'min
d', {'edge_label': 'eleven', 'key': 16, 'edge_order': 2}), ('year', 'pre', {'edge_labe
l': 'old', 'key': 16, 'edge_order': 3}), ('experience', 'child', {'edge_label': 'underta
king', 'key': 5, 'edge_order': 4}), ('child', 'haircut', {'edge_label': 'boyish', 'key':
15, 'edge_order': 1}), ('child', 'expression', {'edge_label': '', 'key': 20, 'edge_orde
r': 1}), ('child', 'yep', {'edge_label': '', 'key': 22, 'edge_order': 1}), ('child', 'gi
rl', {'edge_label': 'sandra', 'key': 24, 'edge_order': 1}), ('thing', 'friend', {'edge_l
abel': '', 'key': 6, 'edge_order': 1}), ('friend', 'something', {'edge_label': '', 'ke
y': 6, 'edge_order': 2}), ('start', 'brief', {'edge_label': '', 'key': 7, 'edge_order':
1}), ('start', 'parent', {'edge_label': '', 'key': 29, 'edge_order': 2}), ('brief', 'int
roduction', {'edge_label': '', 'key': 7, 'edge_order': 2}), ('interruption', 'split',
{'edge_label': '', 'key': 10, 'edge_order': 1}), ('interruption', 'decide', {'edge_labe
l': '', 'key': 13, 'edge_order': 8}), ('group', 'brand', {'edge_label': '', 'key': 10,
'edge_order': 4}), ('group', 'movie', {'edge_label': '', 'key': 13, 'edge_order': 3}),
('group', 'switch', {'edge_label': '', 'key': 12, 'edge_order': 1}), ('group', 'show',
{'edge_label': '', 'key': 13, 'edge_order': 2}), ('brand', 'attendance', {'edge_label':
'', 'key': 13, 'edge_order': 6}), ('brand', 'decide', {'edge_label': 'certain', 'key': 1
3, 'edge_order': 7}), ('movie', 'poster', {'edge_label': '', 'key': 33, 'edge_order':
1}), ('movie', 'return', {'edge_label': '', 'key': 13, 'edge_order': 4}), ('movie', 'wat
ch', {'edge_label': '', 'key': 32, 'edge_order': 1}), ('movie', 'silence', {'edge_labe
l': '', 'key': 32, 'edge_order': 2}), ('poster', 'presentation', {'edge_label': '', 'ke
y': 33, 'edge_order': 2}), ('presentation', 'topographic_point', {'edge_label': '', 'ke
y': 33, 'edge_order': 3}), ('classroom', 'show', {'edge_label': '', 'key': 13, 'edge_ord
er': 1}), ('return', 'attendance', {'edge_label': '', 'key': 13, 'edge_order': 5}), ('at
tendance', 'composure', {'edge_label': '', 'key': 31, 'edge_order': 2}), ('departure',
'list', {'edge_label': '', 'key': 14, 'edge_order': 1}), ('list', 'look', {'edge_label':
'', 'key': 19, 'edge_order': 1}), ('haircut', 'voice', {'edge_label': 'deep', 'key': 15,
'edge_order': 2}), ('voice', 'erosion', {'edge_label': '', 'key': 15, 'edge_order': 3}),
('erosion', 'apparel', {'edge_label': '', 'key': 15, 'edge_order': 4}), ('apparel', 'mal
e_child', {'edge_label': '', 'key': 15, 'edge_order': 5}), ('male_child', 'section', {'e
dge_label': '', 'key': 15, 'edge_order': 6}), ('male_child', 'girl', {'edge_label': '',
'key': 24, 'edge_order': 2}), ('section', 'clothing', {'edge_label': 'large', 'key': 15,
'edge_order': 7}), ('clothing', 'chain', {'edge_label': '', 'key': 15, 'edge_order':
8}), ('chain', 'pipe', {'edge_label': '', 'key': 15, 'edge_order': 9}), ('support', 'min

```
d', {'edge_label': '', 'key': 16, 'edge_order': 1}), ('pre', 'pubescent', {'edge_label':
'', 'key': 16, 'edge_order': 4}), ('pubescent', 'body', {'edge_label': '', 'key': 16, 'e
dge_order': 5}), ('body', 'sexual_activity', {'edge_label': '', 'key': 16, 'edge_order':
6}), ('body', 'manner', {'edge_label': '', 'key': 39, 'edge_order': 4}), ('sexual_activi
ty', 'feature', {'edge_label': 'particular', 'key': 16, 'edge_order': 7}), ('feature',
'eg', {'edge_label': 'dress', 'key': 16, 'edge_order': 8}), ('standard', 'confuse', {'ed
ge_label': '', 'key': 18, 'edge_order': 2}), ('sandra', 'meet', {'edge_label': '', 'ke
y': 29, 'edge_order': 5}), ('sandra', 'humiliate', {'edge_label': 'regretful', 'key': 3
7, 'edge_order': 1}), ('parent', 'mistake', {'edge_label': '', 'key': 29, 'edge_order':
3}), ('mistake', 'meet', {'edge_label': '', 'key': 29, 'edge_order': 4}), ('chitchat',
'gesture', {'edge_label': '', 'key': 36, 'edge_order': 3}), ('humiliate', 'front', {'edg
e_label': '', 'key': 37, 'edge_order': 2}), ('hope', 'continue', {'edge_label': 'health
y', 'key': 38, 'edge_order': 1}), ('manner', 'act', {'edge_label': 'certain', 'key': 39,
'edge_order': 3}), ('anyone', 'dress', {'edge_label': '', 'key': 39, 'edge_order': 1}),
('dress', 'act', {'edge_label': '', 'key': 39, 'edge_order': 2})]
```

Compute node weight

Node weight for each noun is given by a custom weighting equation to deem the importance of each word. The weighting formula will be an adapted formula from [5]

The formula for each node will be given by the following equation. given, i ∈ { words-in-text }

node_weight_i = word_freq_of_i + word_freq_i_in_title + word_freq_i_in_bi_grams

In [15]:

```
import copy

def compute_node_weights(G,word_freq,title,bi_grams_freq,key_words):
    grams = [item for sublist in bi_grams_freq for item in sublist[0]]
    grams_freq = collections.Counter(grams)
    title_freq = collections.Counter(title)
    key_word_freq = collections.Counter(key_words)

    syns_list = []
    title = list(set(title)) #remove_duplicates

    ### stemming (remove -ing, -ly, ...)
    ps = nltk.stem.porter.PorterStemmer()
    cleaned_sentences = [ps.stem(word) for word in title]

    # create dictionary of synonyms
    for word in title:
        synonyms = []
        for syn in wordnet.synsets(word):
            for l in syn.lemmas():
                synonyms.append(l.name())
        syns_list.append(synonyms)

    synonyms_dict = dict(zip(title,syns_list))

    normalized_words = []

    for i in range(len(title)):
        if len(synonyms_dict[title[i]]) != 0:
            title[i] = synonyms_dict[title[i]][0]
            normalized_words.append(title[i])

    title = normalized_words
```

```
    #print(synonyms_dict)

    for node in G.nodes:
        if node in word_freq.keys():
            G.nodes[node]['weight'] += word_freq[node]
        if node in grams_freq.keys():
            G.nodes[node]['weight'] += grams_freq[node]
        if node in title_freq.keys():
            G.nodes[node]['weight'] += title_freq[node]
        if node in key_word_freq:
            G.nodes[node]['weight'] += key_word_freq[node]

    return G

keywords = []
domain_words = []
sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = text_mani
S = copy.deepcopy(sentences_dict)
G = text_graph_construction(S)

G = compute_node_weights(G,word_freq,sentences_dict[0],bi_grams_freq,keywords)

print(list(G.nodes(data=True)))
```

```
[('couple', {'key': 1, 'word': 'couple', 'weight': 2}), ('S#', {}), ('old_age', {'key':
11, 'word': 'old_age', 'weight': 4}), ('#E', {}), ('turn', {'key': 2, 'word': 'turn', 'w
eight': 2}), ('school', {'key': 5, 'word': 'school', 'weight': 6}), ('travel', {'key':
2, 'word': 'travel', 'weight': 4}), ('class', {'key': 37, 'word': 'class', 'weight': 2
2}), ('change', {'key': 14, 'word': 'change', 'weight': 4}), ('name', {'key': 25, 'wor
d': 'name', 'weight': 12}), ('entrance', {'key': 3, 'word': 'entrance', 'weight': 2}),
('grader', {'key': 6, 'word': 'grader', 'weight': 4}), ('life', {'key': 38, 'word': 'lif
e', 'weight': 4}), ('discrimination', {'key': 3, 'word': 'discrimination', 'weight':
2}), ('semitism', {'key': 3, 'word': 'semitism', 'weight': 2}), ('one-third', {'key': 3,
'word': 'one-third', 'weight': 4}), ('spiel', {'key': 3, 'word': 'spiel', 'weight': 2}),
('day', {'key': 41, 'word': 'day', 'weight': 9}), ('student', {'key': 29, 'word': 'stude
nt', 'weight': 6}), ('cancel', {'key': 4, 'word': 'cancel', 'weight': 2}), ('history',
{'key': 4, 'word': 'history', 'weight': 2}), ('survey', {'key': 4, 'word': 'survey', 'we
ight': 2}), ('activity', {'key': 9, 'word': 'activity', 'weight': 4}), ('game', {'key':
4, 'word': 'game', 'weight': 2}), ('stopping_point', {'key': 5, 'word': 'stopping_poin
t', 'weight': 2}), ('year', {'key': 16, 'word': 'year', 'weight': 4}), ('experience',
{'key': 5, 'word': 'experience', 'weight': 2}), ('child', {'key': 24, 'word': 'child',
'weight': 10}), ('thing', {'key': 6, 'word': 'thing', 'weight': 2}), ('friend', {'key':
6, 'word': 'friend', 'weight': 2}), ('something', {'key': 6, 'word': 'something', 'weigh
t': 2}), ('start', {'key': 29, 'word': 'start', 'weight': 6}), ('brief', {'key': 7, 'wor
d': 'brief', 'weight': 2}), ('introduction', {'key': 7, 'word': 'introduction', 'weigh
t': 2}), ('interruption', {'key': 13, 'word': 'interruption', 'weight': 6}), ('split',
{'key': 10, 'word': 'split', 'weight': 2}), ('group', {'key': 13, 'word': 'group', 'weig
ht': 10}), ('brand', {'key': 13, 'word': 'brand', 'weight': 4}), ('movie', {'key': 33,
'word': 'movie', 'weight': 8}), ('poster', {'key': 33, 'word': 'poster', 'weight': 4}),
('presentation', {'key': 33, 'word': 'presentation', 'weight': 4}), ('switch', {'key': 1
2, 'word': 'switch', 'weight': 2}), ('classroom', {'key': 13, 'word': 'classroom', 'weig
ht': 2}), ('show', {'key': 13, 'word': 'show', 'weight': 2}), ('return', {'key': 13, 'wo
rd': 'return', 'weight': 4}), ('attendance', {'key': 31, 'word': 'attendance', 'weight':
4}), ('decide', {'key': 13, 'word': 'decide', 'weight': 2}), ('departure', {'key': 14,
'word': 'departure', 'weight': 2}), ('list', {'key': 19, 'word': 'list', 'weight': 4}),
('haircut', {'key': 15, 'word': 'haircut', 'weight': 2}), ('voice', {'key': 15, 'word':
'voice', 'weight': 2}), ('erosion', {'key': 15, 'word': 'erosion', 'weight': 2}), ('appa
rel', {'key': 15, 'word': 'apparel', 'weight': 2}), ('male_child', {'key': 24, 'word':
'male_child', 'weight': 4}), ('section', {'key': 15, 'word': 'section', 'weight': 2}),
('clothing', {'key': 15, 'word': 'clothing', 'weight': 2}), ('chain', {'key': 15, 'wor
```

```
d': 'chain', 'weight': 2}), ('pipe', {'key': 15, 'word': 'pipe', 'weight': 2}), ('suppor
t', {'key': 16, 'word': 'support', 'weight': 2}), ('mind', {'key': 16, 'word': 'mind',
'weight': 2}), ('pre', {'key': 16, 'word': 'pre', 'weight': 2}), ('pubescent', {'key': 1
6, 'word': 'pubescent', 'weight': 2}), ('body', {'key': 39, 'word': 'body', 'weight':
4}), ('sexual_activity', {'key': 16, 'word': 'sexual_activity', 'weight': 2}), ('featur
e', {'key': 16, 'word': 'feature', 'weight': 2}), ('eg', {'key': 16, 'word': 'eg', 'weig
ht': 2}), ('standard', {'key': 18, 'word': 'standard', 'weight': 2}), ('confuse', {'ke
y': 18, 'word': 'confuse', 'weight': 2}), ('look', {'key': 19, 'word': 'look', 'weight':
2}), ('expression', {'key': 20, 'word': 'expression', 'weight': 2}), ('sandra', {'key':
37, 'word': 'sandra', 'weight': 16}), ('yep', {'key': 22, 'word': 'yep', 'weight': 2}),
('regretful', {'key': 41, 'word': 'regretful', 'weight': 11}), ('girl', {'key': 25, 'wor
d': 'girl', 'weight': 6}), ('parent', {'key': 29, 'word': 'parent', 'weight': 2}), ('mis
take', {'key': 29, 'word': 'mistake', 'weight': 2}), ('meet', {'key': 29, 'word': 'mee
t', 'weight': 2}), ('composure', {'key': 31, 'word': 'composure', 'weight': 2}), ('watc
h', {'key': 32, 'word': 'watch', 'weight': 4}), ('silence', {'key': 32, 'word': 'silenc
e', 'weight': 2}), ('topographic_point', {'key': 33, 'word': 'topographic_point', 'weigh
t': 2}), ('felt', {'key': 34, 'word': 'felt', 'weight': 2}), ('fillet', {'key': 36, 'wor
d': 'fillet', 'weight': 2}), ('chitchat', {'key': 36, 'word': 'chitchat', 'weight': 2}),
('gesture', {'key': 36, 'word': 'gesture', 'weight': 2}), ('humiliate', {'key': 37, 'wor
d': 'humiliate', 'weight': 2}), ('front', {'key': 37, 'word': 'front', 'weight': 2}),
('hope', {'key': 38, 'word': 'hope', 'weight': 2}), ('continue', {'key': 38, 'word': 'co
ntinue', 'weight': 2}), ('manner', {'key': 39, 'word': 'manner', 'weight': 4}), ('anyon
e', {'key': 39, 'word': 'anyone', 'weight': 2}), ('dress', {'key': 39, 'word': 'dress',
'weight': 4}), ('act', {'key': 39, 'word': 'act', 'weight': 2}), ('feel', {'key': 40, 'w
ord': 'feel', 'weight': 4}), ('ignorance', {'key': 42, 'word': 'ignorance', 'weight':
1})]
```

Search the graph using EdgeSumms node-weight algorithm

The algorthim is based off the figure below. It searches the text graph for candidate edges that link between the high-weighted source and destination nodes.

Algorithm 2 Search Graph for Candidate Edges Algorithm

```
Input: G, W, out_criteria
Output: C
Variables: source_node_weight = 0, destination_node_weight = 0
nodes_list = list of nodes and their weights in the text graph where node weight >= 1
average_node_weight = average(nodes_list)
median_node_weight=median(nodes_list)
if median_node_weight > average_node_weight then
  | source_node_weight = median_node_weight
else
  | source_node_weight = average_node_weight
end
for each node in nodes_list
  | if node.weight >= source_node_weight then
  |   | out_nodes = G.get_out_nodes(node)   // get list of destination nodes such that node weight >= 1
  |   | average_weight =get_average_weight(out_nodes) //nodes with weight=0 like "E#" are not counted
  |   | maximum_weight = get_maximum_weight(out_nodes)
  |   | if out_criteria == "avg" then
  |   |   | destination_node_weight = average_weight
  |   | else if out_criteria == "node_avg" then
  |   |   | destination_node_weight = source_node_weight
  |   | else if out_criteria == "max" then
  |   |   | destination_node_weight = maximum_weight
  |   | end
  |   | for each out in out_nodes
  |   |   | if out.weight >= destination_node_weight then
  |   |   |   | C.add(G.get_edge(node,out))  // add edge of the selected source and destination nodes
  |   |   | end
  |   | end
  | end
end
```

Figure 4 Edgesumm search algorithm [5]

This search algorthim has been experimentally developed. However this problem falls under a group of graph problems called node-weighted Steiner Trees [7]. These probelems optimize graph paths for weighted nodes reaching some end-target goal. They iterate through the graph, based on some optimizing heuristic. Steiner Tree's are NP-hard.

'The value of out_criteria could be "avg" (i.e. the default value) for an average value of the destination nodes weights, "node_avg" to use the "source node weight" value for selecting the destination nodes, or "max" for a maximum value of the destination nodes weights' [5].

In [16]:

```
# We will choose our graph search heuristic to be average weight for destination node.
# Thus, out criteria will be based on average node_weight for our implmentation.
#Consider other heuristic it can miss it sometimes.

def graph_search(G,mode):
    source_node_weight = 0
    destination_node_weight = 0
    out_nodes = []

    C = [] #list of edges to be used in summary

    node_list = [(node,G.nodes[node]) for node in G.nodes if len(G.nodes[node]) != 0]

    #get graph weights
    weights = []
```

```
    for node in G.nodes.data("weight"):
        if node[1] != None:
            weights.append(node[1])
    mean = np.mean(weights)
    median = np.median(weights)

    #pick a point where to start search
    if median > mean:
        source_node_weight = median
    else:
        source_node_weight = mean

    #begin graph traversal
    for node in node_list:
        if node[1]['weight'] >= source_node_weight:
            out_nodes = list(nx.neighbors(G,node[0]))
            avg, med = med_and_mean(out_nodes,G,mean,median)
            # find max weight from out_nodes
            if mode == 'max':
                destination_node_weight = max_weight(out_nodes,G)
            if mode == 'avg':
                destination_node_weight = avg
            if mode == 'med':
                destination_node_weight = med
        for out in out_nodes:
            if (len(G.nodes[out])) != 0: #if node is not node #S or #E
                if G.nodes[out]['weight'] >= destination_node_weight: #pick all the nod
                    C.append(list(G.edges(out)))
    return C


def max_weight(out_nodes,G):
    weight = 0
    for node in out_nodes:
        if len(G.nodes[node]) != 0 :
            if G.nodes[node]['weight'] > weight:
                weight = G.nodes[node]['weight']

    return weight

def med_and_mean(out_nodes,G,mean,median):
    weights = []
    for node in out_nodes:
        if len(G.nodes[node]) != 0:
            weights.append(G.nodes[node]['weight'])

    if len(weights) != 0: #covers edge case that the weights list is empty i.e out_node
        mean = np.mean(weights)
        median = np.median(weights)

    return mean, median

keywords = []
domain_words = []
sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = text_mani
S = copy.deepcopy(sentences_dict)
G = text_graph_construction(S)

G = compute_node_weights(G,word_freq,sentences_dict[0],bi_grams_freq,keywords)
```

```
C = graph_search(G,'max')

print(G[C[0][0][1]][C[0][0][1]])
```

```
{'edge_label': 'thirteen', 'key': 2, 'edge_order': 4}
```

In [ ]:

In [17]:
```
### Iteration proccess to deem if summary length has met sentances threshold
```

Create canidate summary text

The canidate summary algorithm is based on the following figure below.

Algorithm 3 Get Candidate Summary Algorithm

Input: C, S, edges_count_threshold=1
Output: candidate_summary
// list of sentences that have edges in the candidate edges list
sentences_list = get_sentences_with_candidate_edges(C, S)
for each s **in** sentences_list
  if get_candidate_edges_count(s, C) ≥ edges_count_threshold **then**
    candidate_summary.add(s)
  end
end

Figure 5 Edgesumm canidate summary algorthim [5].

Algorithm 3 is used to select sentences for the candidate summary. Based on the candidate edges list generated from Algorithm 2 and to narrow the selection of sentences, an edge count threshold is used such that a sentence will be selected if it has candidate edges count greater than or equal to an edges count threshold. But if the threshold value is high, a short sentence with few edges will not be selected this way. Moreover, a whole document or its most important sentences may be short sentences. Therefore, the threshold value is selected to be 1 so a sentence will be selected for the candidate summary if it has at least one candidate edge. This will give a chance to important sentences that are short with fewer edges as well as the longer sentences. **REWRITE**

In [18]:
```
def canidate_summary(G,S,C):
    sentences = []

    for edge_list in C:
        for vertex_pair in edge_list:
            edge = G[vertex_pair[0]][vertex_pair[1]]
            #get the sentence determined by the edge
            #sentences list is a tuple (sentence,key) so we can order the output by sen
            # sentence number = key, which is organized in a zero-base numbering fashio
            if edge['key'] in S:
                sentences.append((S[edge['key']],edge['key']))

    sens = set(sentences)
    return list(sens)
```

```
sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = text_mani
S = copy.deepcopy(sentences_dict)
G = text_graph_construction(S)

G = compute_node_weights(G,word_freq,sentences_dict[0],bi_grams_freq,keywords)

C = graph_search(G,'max')

canidate = canidate_summary(G,segmented_sentences,C)

sorted_canidate = sorted(canidate,key=lambda x: x[1])

sorted_canidate = [x[0] for x in sorted_canidate]

print((sorted_canidate))
print(segmented_sentences)

print(len(sorted_canidate))
print(len(segmented_sentences))
```

```
['my school was named after anne frank and we had a club that i was very active in from
9th grade on which was dedicated to teaching incoming 5th graders about anne franks life
discrimination anti semitism hitler the third reich and that whole spiel', 'basically a
day where the students classes are cancelled and instead we give them an interactive his
tory and social studies class with lots of activities and games', 'this was my last year
at school and i already had a lot of experience doing these project days with the kids',
'i was running the thing with a friend so it was just the two of us and 30 something 5th
graders', 'we start off with a brief introduction and brainstorming what do they know ab
out anne frank and the third reich', 'after the break we split the class into two groups
to make it easier to handle', 'one group watches a short movie about anne frank while th
e other gets a tour through our poster presentation that our student group has been perf
ecting over the years', 'then the groups switch', 'i am in the classroom to show my grou
p the movie and i take attendance to make sure no one decided to run away during break',
'i am going down the list when i come to the name sandra name changed', 'a kid with a bo
yish haircut and a somewhat deeper voice wearing clothes from the boys section at a big
clothing chain in germany pipes up', 'boobs beards', 'i looked down at the list again ma
king sure i had read the name right', 'look back up at the kid', 'i have only ever heard
that as a girls name before', 'the class starts laughing', 'sandra gets really quiet',
'i feel so sorry and stupid', 'we watch the movie in silence', 'after the movie when we
walked down to where the poster presentation took place i apologised to sandra', 'i felt
so incredibly terrible i still do to this day', 'throughout the rest of the day i heard
lots of whispers about sandra', 'i tried to stop them whenever they came up but there wa
s no stopping the 5th grade gossip i had set in motion', 'sandra if you are out there i
am so incredibly sorry for humiliating you in front of your class', 'i hope you are happ
y and healthy and continue to live your life the way you like', 'do not let anyone tell
you you have to dress or act a certain way just because of the body parts you were born
with', 'i am sorry if i made you feel like you were wrong for dressing and acting differ
ently', 'i am sorry for my ignorance']
{0: 'this actually happened a couple of years ago', 1: 'i grew up in germany where i wen
t to a german secondary school that went from 5th to 13th grade we still had 13 grades t
hen they have since changed that', 2: 'my school was named after anne frank and we had a
club that i was very active in from 9th grade on which was dedicated to teaching incomin
g 5th graders about anne franks life discrimination anti semitism hitler the third reich
and that whole spiel', 3: 'basically a day where the students classes are cancelled and
instead we give them an interactive history and social studies class with lots of activi
ties and games', 4: 'this was my last year at school and i already had a lot of experien
ce doing these project days with the kids', 5: 'i was running the thing with a friend so
it was just the two of us and 30 something 5th graders', 6: 'we start off with a brief i
ntroduction and brainstorming what do they know about anne frank and the third reich',
7: 'you would be surprised how much they know', 8: 'anyway after the brainstorming we do
```

```
a few activities and then we take a short break', 9: 'after the break we split the class
into two groups to make it easier to handle', 10: 'one group watches a short movie about
anne frank while the other gets a tour through our poster presentation that our student
group has been perfecting over the years', 11: 'then the groups switch', 12: 'i am in th
e classroom to show my group the movie and i take attendance to make sure no one decided
to run away during break', 13: 'i am going down the list when i come to the name sandra
name changed', 14: 'a kid with a boyish haircut and a somewhat deeper voice wearing clot
hes from the boys section at a big clothing chain in germany pipes up', 15: 'now keep in
mind these are all 11 year olds they are all pre pubescent their bodies are not yet show
ing any sex specific features one would be able to see while they are fully clothed eg',
16: 'boobs beards', 17: 'this being a 5th grade in the rather conservative for german st
andards bavaria i was confused', 18: 'i looked down at the list again making sure i had
read the name right', 19: 'look back up at the kid', 20: 'me you are sandra', 21: 'kid y
ep', 22: 'me oh sorry', 23: 'thinking the kid must be from somewhere where sandra is bot
h a girls and boys name where are you from', 24: 'i have only ever heard that as a girls
name before', 25: 'the class starts laughing', 26: 'sandra gets really quiet', 27: 'i am
a girl she says', 28: 'some of the other students start saying that their parents made t
he same mistake when they met sandra', 29: 'i feel so sorry and stupid', 30: 'i get the
class to calm down and finish taking attendance', 31: 'we watch the movie in silence', 3
2: 'after the movie when we walked down to where the poster presentation took place i ap
ologised to sandra', 33: 'i felt so incredibly terrible i still do to this day', 34: 'th
roughout the rest of the day i heard lots of whispers about sandra', 35: 'i tried to sto
p them whenever they came up but there was no stopping the 5th grade gossip i had set in
motion', 36: 'sandra if you are out there i am so incredibly sorry for humiliating you i
n front of your class', 37: 'i hope you are happy and healthy and continue to live your
life the way you like', 38: 'do not let anyone tell you you have to dress or act a certa
in way just because of the body parts you were born with', 39: 'i am sorry if i made you
feel like you were wrong for dressing and acting differently', 40: 'i am sorry i probabl
y made that day hell for you', 41: 'i am sorry for my ignorance'}
28
42
```

Just from a single iteration of the algorithm the canidate summary is almost half the amount of sentences from the given text

Proccess iteration to reduce canidate summary text to close to threshold

In this implmentation there will be no hard requirments on meeting the thershold number. Instead the summary generation proccess will be iterated over **SOME GRAPH METRIC**

In [19]:

```
from nltk.tokenize.treebank import TreebankWordDetokenizer

def build_text_summary(text,title,sentence_threshold,key_words,mode):
    sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = text_
    S = copy.deepcopy(sentences_dict)
    G = text_graph_construction(S)

    G = compute_node_weights(G,word_freq,sentences_dict[0],bi_grams_freq,keywords)

    C = graph_search(G,mode)

    canidate = canidate_summary(G,segmented_sentences,C)

    sorted_canidate = sorted(canidate,key=lambda x: x[1])

    sorted_canidate = [x[0] for x in sorted_canidate]
```

```
    text_canidate = ". ".join(sorted_canidate)+"."

    #according to paper the text might not actually converge to the sentence threshold
    #so after a set number of iteration just return the document

    iterations = 10 * len(sentences_dict)
    counter = 0

    while len(sorted_canidate) > sentence_threshold or counter > iterations:
        sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = t
        S = copy.deepcopy(sentences_dict)
        G = text_graph_construction(S)

        G = compute_node_weights(G,word_freq,sentences_dict[0],bi_grams_freq,keywords)

        C = graph_search(G,mode)

        canidate = canidate_summary(G,segmented_sentences,C)

        sorted_canidate = sorted(canidate,key=lambda x: x[1])

        sorted_canidate = [x[0] for x in sorted_canidate]

        text_canidate = ". ".join(sorted_canidate)+"."
        counter += 1

    return text_canidate

key_words = []
summary = build_text_summary(cleaned_data[6][0],cleaned_data[6][1],5,key_words,'med')
print(summary)
print('text')
print(cleaned_data[6][0],cleaned_data[6][1],cleaned_data[6][2])

#print(max(cleaned_data, key = lambda i : len(i[0]))[0])

### stemming (remove -ing, -ly, ...)
# ps = nltk.stem.porter.PorterStemmer()
# ps.stem('pepperminty')
```

```
steam billowing out the crack between the two plates. i lift off the top plate and let i
t cool down a but i touch the french fry and it is warm but a tad bit cold i feel around
and flip them with my fingers genius. and then i reach the center of the plate. freeze f
rame have you ever microwaved something like chicken nuggets and the nuggets near the ed
ge of the plate are colder than the ones in the middle. well this is exactly what happen
ed but a thousand times worse.
text
so, my oven is broken...and i'm out of cooking oil, but i want some french fries. so i s
earch the internet and find that if you cover french fries with something, either napkin
s or another plate and microwave them, they turn out ok. so i tried it, i took to plate
s, smushed em together and put the french fries in the microwave. (don't try this at hom
e)
 half way through the five minute cooking cycle i pull the plate out. steam billowing ou
t the crack between the two plates. i lift off the top plate and let it cool down a but,
i touch the french fry and it's warm but a tad bit cold, i feel around and flip them wit
h my fingers (genius!) and then i reach the center of the plate.
 freeze frame, have you ever microwaved something, like chicken nuggets, and the nuggets
near the edge of the plate are colder than the ones in the middle? well this is exactly
what happened, but a thousand times worse. since the fries have been blanched, frozen, b
agged, and then put into my freezer, that means there's still some oil in the fries, and
```

```
the microwave brought out that oil and boiled it,
 so i was poking around the fries when my finger squishes into a really soft one, right
into the hell-fire center, where broiling hot potato smushes under my finger nail, i jum
p back in shock and flail around my hand whispering "ow, stop, ow, stop, ow, stop" i cov
er the plate, start the microwave, and walk away, suddenly, this burning sensation just
erupts over the tip of my finger, full on pain train. i step into the bathroom and run i
t under cold water, only to find out. we're out of burn cream!
so now i'm sitting at my desk, with first degree burns on my index finger which i stuck
in a bag of frozen corn
as i type with one hand.
update: i put aloe vera on the end of my finger and it feels like little tiny snow angel
s grinding up against my finger, so gud ['tifu', 'by', 'trying', 'to', 'microwave', 'fre
nch', 'fries']  tried to microwave french fries and stuck my finger into a franken-fuck
of pain causing first degree burns
```

Some tests :)

This algorthims supposedly has preformed quite well compared to other ATS' according to [5] and the testing metrics they use to for the sake of simplicity lets look at some examples and see if they are hummanly deemed as a solid summary.

Lets look at what happens when we summarize the largest document and the smallest document and see how they preform under differnt graph searching heuristics.

In [20]:

```
max_summary = max(cleaned_data, key = lambda i : len(i[0]))

sorted_by_summary_len = sorted(cleaned_data, key = lambda i : len(i[0]))

#dataset has issue where there maybe
# a tl;dr (to long didn't read -- the authors summary) but no body text
# so we just take the first time a summary is longer than a tl;dr




for summary in sorted_by_summary_len:
    tokenizer = PunktSentenceTokenizer()
    tokenizer.train(summary[0])
    text = tokenizer.tokenize(summary[0])
    tldr = tokenizer.tokenize(summary[2])
    title = summary[1]
    if (len(text) > len(tldr)) and (len(text) > len(title)):
        min_summary = summary
        break

print(len(max_summary[0]))
print(len(min_summary[0]))
```

```
31476
207
```

In [21]:

```
summary = build_text_summary(max_summary[0],max_summary[1],5,key_words,'med')

print(summary)
print()
print('TEXT')
print(max_summary[0],max_summary[1],max_summary[2])
```

if i got a fucking penny for every time i heard the phrase oh my god i am so fucking dru nk after half of a beer at sea level i would be a millionaire. my english teacher this y ear warns us all the time about doing something stupid for susie q but a lot of the guys laugh it off. they think they have never been somewhat forced to do something for of a g irl but one of those laughing guys drove a girl while he was drunk so he could get her a burger from in n out. it is like what the fuck were you thinking. i have never driven dr unk and i do not want to.

TEXT
so this happened to me sunday night, and instead of locking up my computer like i should have, i did it again. for some reason when i am on adderall (only taken twice) i just wa nt to talk and express my ideas. what better environment to do this than on a blank goog le doc. for the second night now, i have written nine pages of just my ideas and thought s about life. i don't know why i did this, or why i let it happen a second time. to be c lear, the two seven hour periods would add up to fourteen hours of useless writing with eighteen pages of my thoughts. i am in my senior year of high school by the way and i ha ve studied for ap calculus exactly zero minutes and it is ten o'clock. kids are posting statuses on facebook about getting into yale and my regular decision-ass is writing dissertations on politics and social attitudes towards marijuana.
people wanted the full thing so here is the first one i wrote
preface
i have lived a short life. at this exact moment, i am seventeen years old. i have experi enced a lot, but at the same time i have such a long way to go. the idea of writing a bo ok always got me upset. a lot of people that know me can attest to the fact that i have a very vivid, creative imagination. i like to talk a lot, and most of the time it is com plete bullshit.
by the way, this book is going to have some naughty language in it. sorry.
anyways, in my senior year of highschool i took a semester class on modern american nove ls. of all of the books i read for the class, i discovered two writers. f. scott fitzger ald and tim o’brien. sure i knew about fitzgerald, but i never had read the great gatsby and that was a nice read. that was bullshit, never read the book. i was suppose to, but i said fuck it and went straight to sparknotes the night before the test. i liked fitzge rald because he was documenting life in the 20s. it was all fiction, but it gives reader s a taste for the time period. tim o'brien's work is just phenomenal. the war genre was not what drew me to him, it was his voice. his writing style is personal. o'brien writes to the reader and his way of storytelling is my favorite.
given all of those influences, i realized that i wanted to tell my story. i was born in 1998, and my lifetime will definitely be very well documented. the internet is here, we have websites and emails. donald trump is running for president. drones are starting to become more common. there’s phones made out of glass, it’s fucking crazy.
but if it's going to be so well documented, then why write about your story? when we loo k at history, which i do a lot (you will learn about that later in the book), historians are essentially the final say about what really happened. sure, the victor writes histor y, but as time goes on people can gather enough evidence to really figure out what happe ned. and historians aren’t going to learn about 2015 from fox news, vine, or a subreddi t. so to future historians, you are welcome. maybe you will realize that we are a bunch of idiots right now.
also, i wanted to write a book just so i can say that i wrote a book. if this gets publi shed and makes some money, future me you are welcome. i doubt i will write a book-length document, but fuck it let's go. so with that long and personal preface, let’s begin.
about me
before i start rambling about the world around me and what i think about it, i am going to give you the reader some context about myself and where i come from.
i am, at this exact moment of writing, seventeen years old. i go to [removed] and i am a pplying to colleges. i live in the greater los angeles area, specifically [removed], wit h my father, mother, younger brother, and labrador retriever. i would assume we are midd le class, my parents can afford for me to drive my own car. i have, up to this point, al ways attended private school. i am caucasian.
my mom went to san diego state university and majored in communications. my dad went to

usc for electrical engineering, but dropped out after three semesters. both of my parent
s are from glendale california. my mom’s parents are foreigners from peru and argentina,
but they are both caucasian because their families migrated from france and russia.
my dad is italian, to an extent, and the ‘[removed]’ family is very large. our lineage c
an be traced back to my great-great grandfather who was an orphan. the only reason we ha
ve the [removed] last name, a very common last name, is because he picked it himself. i
would have gone with something more mafia-scary like correleone, but it is what it is.
my hobbies include reading, surfing, snowboarding, and sleeping. i am five foot nine in
height with brown hair. a jewish classmate has described my nose as being ‘a right trian
gle’, but i am russian orthodox. i do not speak russian and rarely attend services. my r
eligion is kind of like the mole on my back, i scratch at it when i want to. i am not de
dicating a chapter in this book to religion because i am not a big ‘organized religion’
kind of guy, but i like to think that god exists.
that’s really all i am going to write about my background for now. i would like to think
that the more you read about me in the following pages the more you will know about me a
nd my personality.
reading
books are fucking fantastic. ever since i could read, i was reading books. i wanted to r
ead harder, and more challenging books. i wanted to one-up other kids with my ‘words per
week’ score and with my completed books list. one of my friends, thomas, was telling me
during break about how his dad was reading this great book to him called harry potter. f
irst grade me was like fuck reading with parents, i’ll read that shit to myself. harry p
otter really was what started my obsession of reading. i finished all of the published b
ooks by fourth grade and would continue to read the new ones until their eventual end.
now at this point i was into reading book series’ because i like the continuation of sto
ries. my own personal goals were to finish the narnia books by the end of the third grad
e. i knew that with my speed i could do it, but there was just one problem. my parents.
they knew that i liked to read. i liked it so much that i did it instead of homework. no
w to parents who are reading this, i know i am the best kid ever. other people my age ar
e probably laughing at how much of a fiction nerd i was. kids in grade school want to pl
ay jump rope and kickball, not sit under a tree and read.
my parents didn’t know what to do at first, how do you punish a young child for free rea
ding? they would check in on me constantly, looking for an open book. i combated this sp
ying by locking myself in the bathroom and hiding my book under my shirt. my grandma wan
ted to take me to the hospital one time because i had been on the toilet for three hour
s. i needed more reading time, but at the same time i needed to get good grades. for tha
t class, i was struggling in math. i was supposed to be learning about the times tables,
but my own ingenuity prevented that by hiding books underneath my desk. i had to protect
my books, so when math became a problem at home, i decided to change my grades.
the times-table quiz sheets were passed out in class all the time for ‘at home practic
e’. you had to start with one and go all the way to twelve, multiplying each level numbe
r with numbers one through twelve. so what i did, and understand i was in the third grad
e, was make copies of the empty tables. i then would use a calculator to fill them out f
or each number. then i would take those to school, one by one, and switch the completed
pages for the uncompleted pages that the teacher would distribute during testing time. i
t was flawless, we had those old-school desks with the compartment underneath so it was
pretty easy to make the switch. i was the second kid to pass all twelve levels, even tho
ugh i was behind, and i have no regrets. afterall, i got to finish the seven narnia book
s.
school, pressure, and drugs
ok well i thought the next best place to start is school, specifically high school. like
i mentioned earlier, i go to [removed]. it is an all boys, jesuit-run, school that has b
een around since 1865. it is not that easy to get into, and it is considered to be one o
f the tougher schools in southern california. that first semester sealed my academic fat
e. i got two c’s in english and physics (yeah, they made freshman take physics) and that
did not set me up well. freshman, except under certain conditions, are not supposed to t
ake honors or ap classes. your first semester grades give the different academic departm
ents the required information for your access to ‘restricted’ classes for the next year.
then if you do good in those, they let you take more in your junior year, and all the wh
ile your unweighted and weighted gpa goes through the roof.

now because of that disastrous semester, i did not get to take any gpa booster classes u ntil my junior year. and in that junior year i only got to take one, ap us history, and mind you i had to hassle the social science department head for that seat. the class was hard and i think i had the smartest kids in my grade in that class. two kids in that cla ss are going to brown, another to harvard, and one who just got into stanford. these kid s were geniuses. i felt so out of place, since i had practically killed it among the ‘st upid’ kids in my sophomore year.

for that entire year i was so focused on the ap exam. i was reading about the test, the times you get, and the essays you have to write. i was scared. i didn’t want to fail. if you don’t get a three on the exam, then you don’t get college credit, and if you don’t g et college credit what’s the point of putting yourself through hell.

so i sought out supplements. you hear about prescription drugs like adderall and focalin helping kids study, but the side effects listed on the internet can be daunting. lack of sleep, no hunger, buzzing sensation around your skin were just a few, but i wanted to tr y it anyways. there was a kid in my english class who was prescribed for adderall for hi s adhd. this was about a week before the exam. i had recently gotten a sixty-four percen t on our practice exam. anyways, this kid doesn’t want to take his pills because he said that he wanted to eat. now i am a small, skinny guy and i don’t eat a lot anyways so whe n he offered them to me the fear of skipping lunch was at the back of my mind. i went to the bathroom and used sink water to digest them. i digested the pills at around 9:30 am, it was the first class of the day. that wednesday at school was probably my best day of school so far, and the only thing that will rival it might be graduation, but we haven't gotten there yet.

so i kind of start to feel a little edgy, i’m starring in certain directions for longer periods of time, my eyes are focused, and i’m not super talkative like i usually am. i w as dialed the fuck in. i was listening to everything the teacher was saying and i was ac tually absorbing the information. i don’t know what the dosage was, but i think it was p retty high because since then i have taken adderall a few times and never has it been so strong. skip a few periods ahead and i’m in apush. we get our practice tests back and we are going to go over the answers. now at this point i am the sharpest i have ever been. we are going through every question, one by one, and i was like a fucking magician. i kn ew every answer.

sidenote, so the new apush test had a change in the multiple choice section. instead of asking specific questions, they were going to give a historical passage that you would h ave to interpret and then answer several questions about. they could be from martin luth er king jr. or they could be from president lincoln.

i was reading each one at very fast pace and answering each question correctly. it was i f this new mental state was allowing my mind to work more efficiently. after class i fel t very confident about the test next week, but my best school day was far from over. add erall, i guess to my understanding, is suppose to take the jumbled and energetic mind of a kid and make it more efficient and focused. my classes were basically over because my next period was pre-calculus but we had a sub so that meant studyhall. so after apush, w as lunch. again, i was not hungry because of the drugs. i decided to use that time effic iently.

the lists for ap classes for senior year had been posted, and much to my dismay i had be en accepted to only two. ap government and ap studio art. i was not happy with that. i t hen proceeded to go to the science and social science department chair heads and practic ally badger them about my courses for next year. they all said the same thing, “your gpa is too low”. that didn’t stop me though, because after twenty minutes i got accepted int o ap physics, ap european history, and ap human geography. i didn’t want to take ap gov. because i thought the entire class would be concentrated on concepts that i could learn about on house of cards. i dropped studio art because, quite frankly, i wasn’t big on ta king pictures. even for ap credit. i even had time to email my math teacher about bumpin g me up a percent to an a minus so i could get into ap calculus.

those pills controlled my brain until 1:30 the next morning and it was not until then th at i was able to fall asleep. i was tossing and turning for hours, thinking about how wo nderful it was, but in the back of my mind i knew i was getting addicted. i didn’t want to go down that path, so i told myself i wasn’t going to try and get a legal prescriptio n.

so now here i am in my senior year, taking four incredibly difficult ap classes and all

the while trying to finish college applications. people, teachers and students, have told me that i am crazy because of my own academic boundaries. in hindsight it was not the greatest idea, but it hasn't been to bad. i really did it for college. my transcripts show a struggling student who has challenged himself as much as he could throughout his four years. will it get me into a dream school like usc who has an average acceptance rate of students with 3.9s? no, but i will frame that letter of rejection and i will take it with me to every job i have ever had. that's another fucked up system, the college application process. so the world today wants me to not have a perfect body or perfect life because perfect is overrated, but i need a perfect gpa to get into most of the good colleges?

high school girls and alcohol

i have never had a girlfriend, gone to an invite dance like formal or prom, or even really 'been' with a girl. going to an all-boys school has definitely added to fear of women. and it's not like i am gay or anything, not saying anything is wrong with being gay, but women are scary. to be brutally honest, women are smarter than men. they are a cunning and ruthless sex who can make men do a lot of stupid shit. i'm not trying to be sexist here, but women are the bane of a man's existence. an attractive girl can get a guy to go out of his way for her, and it is especially easy for guys to do this at my age. we are at the height of our sexual prime, there is a lot of testosterone going around. but since i would consider myself to be somewhat of an introvert when it comes to girls, i am safe most of the time from their control.

high school girls can be super annoying. their voices are super high, they think they're the shit, and they can complain a lot. but if a guy might has a slight chance of hooking up with her, then he his just oblivious to all of that. i have seen the fake drunk girl at parties numerous times. if i got a fucking penny for every time i heard the phrase "oh my god, i am so fucking drunk!" after half of a beer at sea level, i would be a millionaire.

my english teacher this year warns us all the time about doing something stupid for susie q, but a lot of the guys laugh it off. they think they've never been somewhat forced to do something for of a girl, but one of those laughing guys drove a girl while he was drunk so he could get her a burger from in n out. it's like what the fuck were you thinking? i have never driven drunk, and i don't want to. i don't need to anyways because of apps like uber and lyft.

alcohol is a drug, no matter what you think. so is tobacco and marijuana. these drugs are fun. people in general like to get fucked up. there is almost a universal desire among all creature to get fucked up. jaguars chew on certain plants to get high, birds purposefully eat fermented fruits, humans make their own drugs, it's just the way it is. it's like sex, everyone has a desire for sex. you get dopamine from sex, so sex is technically a way of getting a little high.

to be clear i do not condone the use of drugs or alcohol, so all the crazy conservative parents who think that i am the spawn of satan you can stop updating your hate blog and save those stamps for your christmas cards.

in california, it is pretty easy to obtain weed. i visited colorado and washington when they were legalized and i still see more dispensaries over here. you have to be eighteen to get a medical card, after the joke of a doctor's consent, and you can legally purchase marijuana. i have been told that you spend more time filling out your information and waiting for the doctor than you do in his office.

doctor: "hello, what's wrong with you?"

patient: "my back hurts."

doctor: "that's unfortunate." signs name

doctor: "have a good day."

it is also very potent. the quality of marijuana has increased significantly and its effects have gotten stronger and last longer. it's a billion dollar industry, and i like to think that it is our 1970s-like computer oppurtunity. to be more clear, if steve jobs and bill gates were born in 1996, they would have dropped out of harvard and come down to southern california instead of silicon valley to revolutionize the marijuana industry.

we're not done yet with recreational habits like drinking and smoking, but we'll come back to it after the most popular chill spot is addressed. college.

college and partying

now college is a fantastic place, or so i have heard. you can get a degree in anything y

ou want, if you get into those programs. you can have a starting salary of a hundred gra nd after graduation, but don’t forget about those college loans you took out. this excer pt will be relatively short because i lack a lot of experience with the college scene, b ut i do have one story to tell.

so i was at boulder, colorado to check out the school with my friend and my parents. i h ave some friends who go there, so i took my friend to go visit them. of course we partoo k in activities related to college, but i had never been exposed to this level of intens ity before. these kids could drink like nothing i had ever seen before. that might be a bit of a lie, but the density of how many people could drink that much was astonishing. and we were in fucking colorado during november and it was twenty degrees outside and th ese kids are wearing t-shirt, some with a light sweater. meanwhile i am back and forth t o my friend’s house, which he shares with seven other guys, grabbing extra layers.

and the party was outside! why not have parties indoors, especially there. this student friend of mine who took my other friend and i under his wing was a twenty-one year old s enior. i guess you could say we were very intimidated by him and his friends because for starters they were all practically four years older than us. we were children who stood out like a sore thumb. also, one of his roommates was a very violent drunk. he threw a c hair across the living room into a door and then continued to tried to break the door. a ll the while his buddies cheered him on. it was very scary. i had always assumed i would want to rush a fraternity, but after seeing that i was very unsure about my ability to h ang. they invited us to come back the next day to witness them do the great american cha llenge. i’m not going to tell you what the great american challenge is, so refer to urba n dictionary. it involves a lot of things including a thousand piece jigsaw puzzle and a n extra large pizza, but i honestly don’t want you make anyone queasy.

college seems like a cool place though, and i hope i have a good experience. after all i t’s only for four years, and high school flew by like a bullet. life moves fast, that’s the only thing i know for sure about this world.

kids and parties

now i am not a big partier. i go to a lot of functions, but i’m not the guy who takes ei ght shots and i’m not volunteering for keg stands. i do not really like alcohol all that much, i think it's a literal poison that makes you feel cool at night and sick in the mo rning. kids today like to drink, obviously, and it is easier now for kids to find partie s. everything is online. facebook events seem to be the most popular option. other than that people get contacted via text message and they tell a few people and it spreads lik e the flu. some parties have lists, others don’t. some are big, others small. some get s hutdown by police, some don’t. some are in the extravagant homes of brentwood, other’s a t a beach house in malibu.

because my school draws kids from all over los angeles, there are party opportunities ev erywhere. it is kind of like a cultural lesson. lifestyles of people in bel aire are sig nificantly different than that of koreatown. the pacific palisades gets their police fro m culver city, so it is easy to keep a party going longer there. there are checkpoints o n pch at night. the cops in manhattan beach have nothing really going on so all they do at night is drive drunk kids home. the streets in hollywood are full of drunk tourists a t night, it almost resembles that of las vegas but without the humid air and casinos.

the internet has not helped with the fight against fake i.d.s. alcohol is therefore easy to obtain, which i guess has always been true except for the era of prohibition. i learn ed from my dad that when he was in high school it was fairly easy to get a ‘real’ fake i.d.

so way back when you apparently could access public records to the extent that you could walk away with a copy of someone’s birth certificate. so these high school kids would go to a cemetery, find a dead child essentially who would be twenty-one, and they would go get a copy of his or her birth certificate. the dmv at the time didn’t have death times recorded in their systems, and since a birth certificate is all you really need, you han d it over and get your picture taken. sure it’s a different name but it’s also a real i. d. today that would never work, but kids have always been crafty with this kind of stuf f. again, everyone wants to get fucked up or at least pretend that they’re getting fucke d up.

the common recreational party activities has changed though, and unfortunately for the w orst. beer pong is still around, the use of alcohol and party drugs still exist, but the science of the twenty-first century has given us the vaporizer. originally built for pe

ople who are trying to quit smoking cigarettes, the vaporizer, or vape, can be used to h
elp them reduce their use of nicotine over time without the harmful effects of tobacco.
kids though, who still think smoking is cool despite the known risks, have also transiti
oned to the use of the vape. now we have these retards on social media doing smoke trick
s, all the while they are getting hooked on nicotine. i know i should care about their h
ealth, but i really don’t. if you’re one of those kids who paid two hundred dollars for
a ‘mod’ and you won’t shut-up about your enhanced atomizer or whatever bullshit engineer
ing parts are involved, i fucking hate you.
i really feel bad for the anti-tobacco guys who worked so hard to get the public to say
fuck off to the big tobacco companies, all their hard work has just gotten us to an even
worse place. sure the vapists are not harming themselves, at least as far we know today,
but it is so dumb that words in the english dictionary cannot give it an adequate name.
and i don’t really ‘fucking hate’ the vapists, but you guys look really fucking stupid.
we asked for flying cars and hover boards, but this is what we got instead.
southern california
oh man, my favorite place in america. i’ve never lived anywhere else, but even if i had
i think that it would still be my favorite place. to understand a complex state like cal
ifornia, you first have to look at its size. it’s not the biggest state in the union, bu
t it is big. there is a a lot of diversity between the different ‘regions’. first off yo
u have the best part, southern california, if any norcal kids say that i’m wrong then te
ll them that they’re indeed the ones who are “hella wrong.” socal is densely populated a
nd has a lot of land. the temperatures are hot, but not arizona hot. arizona hot is too
hot. southern california has all of the cool stuff like the movie industry, the famous b
eaches, the famous streets, the famous people, great mexican food, six flags, and disney
land. central california is cool too, but a lot of the inland area is used for agricultu
re. the action for them happens closer to the coast, like the cities along monterey bay.
now when we look at the state, you can notice that the ‘northern people’ like people fro
m san francisco and sacramento are kind of in the middle of the state. but that’s what w
e consider to be norcal. now what is north of norcal? probably sasquatch tribes and hipp
y city states that are yet to be discovered.
the main thing to take away is that socal is the best. now i will brag and brag about my
‘home’, but i hate tourists. they drive me crazy. i hate seeing rental cars on the 405 a
nd the 101 freeways driving the speed limit in the fast lanes, i just want them to go ba
ck home. the people who feel obligated to slow down on mulholland drive to take a pictur
e of the scenic san fernando valley piss me off too. i blast hard rock like rage against
the machine when i am pulled up next to one of those tourist celebrity tour vans. just l
eave those people alone, they don’t want you taking pictures of their homes.
for some reason, southern california produces the most water-polo players. i guess water
-polo on the west coast is like lacross on the east coast, it’s really big on one side b
ut not on the other. also, statistically speaking the weather is warmer and i can only a
ssume that more pools are built than ice rinks in california. people also like to come h
ere for our schools. if i was an international student, i would definitely work my ass o
ff to get into a school like usc or ucla just because of where they are. kids in this co
untry also want to come to california, it’s a unique place. the biggest downside is that
we are a part of america and therefore cannot legally bar our own citizens from enterin
g.
celebrities
i really hate the idea of a celebrity. i would never want to be a celebrity. people chas
e you with a camera and try to get you to talk or touch them so you can get them paid. i
t’s ridiculous. they are normal people who happen to have a lot of money. when they make
mistakes they get ridiculed by everyone and they are idolized as invincible by super-fan
s. i live near the studios, specifically disney and warner brothers. my neighborhood has
a lot of celebrities in it because of its proximity to these studios, so it’s understand
able that they want to be close. everytime i drive past someone's house, whether it be m
iley cyrus or steve carell, there are people outside waiting. it is absolutely ridiculou
s.
if you thought i like to talk straight bullshit, then go meet someone who works at a stu
dio. it is a cesspool for bullshit. i have been around that world my entire life, and yo
u meet a lot of interesting people, but they are full of shit. it is an industry of liar
s and gutter politics. it seems that the only sane people are the creative animators. a

nything outside of those studio offices is bullshit. marketing, advertising, pr, publici
sts, executives, talent, directors, producers, the writers especially, all bullshit.
these are crafty people, a breed of lawless integrity. everyone thinks they have the nex
t movie idea, or the next big screenplay. it’s a disease. people move from all over the
country to try and be an actress or actor and they get nowhere because it is not what th
ey thought it would be. it is not easy to make it in this industry, you have to have ice
in your veins and a heart of stone. people get hurt, people get rich. it is a backwards
psychotic world, but everyone is just looking out for two things, themselves and their w
allets. my heart goes out to great, young creative artists who think they can be the nex
t wes anderson. they instead are forced to direct porn to make ends meet. it may sound a
little fucked up but if the studios won’t take you there’s always porn. sex sells, tha
t’s the oldest trick in marketing. don’t ever forget that.
the media and politics
this will be great for adults in this country who are confused about the youth’s politic
al agenda. except for the pc, super informative students, no one has a single fucking id
ea about what is going on in politics. the modern day career politician is not what our
founding fathers envisioned, you’re not suppose to do stupid shit like get bought out by
corporations or get caught cheating on your wife. what i said about us not knowing wha
t’s going on in politics is true, but people think they know. turn on fox news, watch te
n minutes, now you know why donald trump is the best. watch cnn for ten minutes, shit tr
ump sucks. sanders is the call. the political media is just fucking stupid, like the reg
ular social media.
we can now literally see the stupidity of people on our devices. from twitter to vine, p
eople doing and saying stupid shit is everywhere. people will do anything for a like, fa
vorite, smile, revine, rehash, and all of the other modes of approval that exist. like t
he old saying goes, everyone is a critic. everyone thinks they know the truth, but reall
y they just believe in their opinion. take that one to the dinner table. my dad is a dev
out republican. i don’t hate for respect him for it, it just is what it is. he is a repu
blican and i am undecided. he can tell me what he thinks all he wants, but i will not be
broken by his convincing tone. same goes for liberals. i hate people who are from both s
ides, but a lot more of them are liberal. the liberal teachers and classmates i have had
think that if they talk really loud and authoritatively then their opinion becomes true.
they throw facts out about bush, guns, and more bush. and then i am politically aware en
ough to decide for myself that obama isn’t really anything special. but then again, like
trump, you can’t judge a party by its popular leader. i think i should just make a new p
arty for people who are equally confused. i think that gun laws and all of that other ho
rse shit should be determined state by state. we are too big of a country to decide for
everyone, so break up the decision making. the media then swoops into this big mess on a
falcon and fucks up your brain even more with facts, speakers, and breaking news headlin
es. it’s like you want to have an opinion, but you don’t want to offend or get into a po
litical debate with your friend. that’s my two cents anyways. just blatantly confused, a
nd concerned. i can’t vote yet, so why even bring it up for discussion. ['tifu', 'by',
'taking', 'adderall', 'for', 'finals', 'and', 'free', 'wrote', 'almost', '0', 'words',
'of', 'nothing'] ---i took pills and i wrote a bunch of stupid shit

In [22]:
```
summary = build_text_summary(min_summary[0],min_summary[1],5,key_words,'avg')

print(summary)
print()
print('TEXT')
print(min_summary[0],min_summary[1],min_summary[2])
```

simple story. i went to take a snapchat and found a glob that was not rubbed in by my l
ip.

TEXT
i'm on the bus on the way to my high school and this just happened.
simple story. i put lotion on my face and got on the bus. i went to take a snapchat and

```
found a glob that wasn't rubbed in by my lip. fml. ['tifu', 'by', 'using', 'lotion']  fa
ke jizz on my mouth
```

An extension of the system: the researchers dream library

Imagine you are a research who is searching through a online library of text documents. You are researching some eccentricity about this strange subreddit you have found r/tifu. You want to do a quick survey of these reddit posts, i.e categorize them, a get a general understanding if posts have similarities ect. One problem though. The number of posts on this reddit is massive. Way to big for a single researcher to search through all, find similar posts and complie summeries of them.

Fear not! There is a solution! The Magic Library.

The magic library

A based on a given document title return all the similar documents and provide document summaries of the returned documents.

To get similar return similar documents based on an implmentation of LSH [8]. If you would like to read more about lsh, you can find them here [9] [10]. This system is very fast for returning similar documents with good accuracy

Build the Library

Our example library will contain 100 documents. This is because the shingling matrix and signature matrix can get very big and we have no off disc storage to hold such a large matrix.

In [59]:

```
key_words = []

docs = []

num_of_docs = 0
doc_id = 0


for data in cleaned_data:
    if num_of_docs == 100:
        break

    if len(data[0]) > 0: #not an empty document body
        txt = []
        sentences_dict, word_freq, bi_grams_freq, bi_gram_tags, segmented_sentences = t
        for key in segmented_sentences:
            txt.append(segmented_sentences[key])
        txt = " ".join(txt)

        title = "_".join(data[1])
        docs.append((txt,title,doc_id))
        num_of_docs += 1
```

```
    doc_id += 1

print(docs[0])

# data =

# lsh_model = LSH(data)
# num_of_random_vectors = 15
# lsh_model.train(num_of_random_vectors)

# #find the 5 nearest neighbors of data[1] while searching in 10 buckets
# lsh_model.query(data[1,:], 5, 10
```

```
('this actually happened a couple of years ago i grew up in germany where i went to a ge
rman secondary school that went from 5th to 13th grade we still had 13 grades then they
have since changed that my school was named after anne frank and we had a club that i wa
s very active in from 9th grade on which was dedicated to teaching incoming 5th graders
about anne franks life discrimination anti semitism hitler the third reich and that whol
e spiel basically a day where the students classes are cancelled and instead we give the
m an interactive history and social studies class with lots of activities and games this
was my last year at school and i already had a lot of experience doing these project day
s with the kids i was running the thing with a friend so it was just the two of us and 3
0 something 5th graders we start off with a brief introduction and brainstorming what do
they know about anne frank and the third reich you would be surprised how much they know
anyway after the brainstorming we do a few activities and then we take a short break aft
er the break we split the class into two groups to make it easier to handle one group wa
tches a short movie about anne frank while the other gets a tour through our poster pres
entation that our student group has been perfecting over the years then the groups switc
h i am in the classroom to show my group the movie and i take attendance to make sure no
one decided to run away during break i am going down the list when i come to the name sa
ndra name changed a kid with a boyish haircut and a somewhat deeper voice wearing clothe
s from the boys section at a big clothing chain in germany pipes up now keep in mind the
se are all 11 year olds they are all pre pubescent their bodies are not yet showing any
sex specific features one would be able to see while they are fully clothed eg boobs bea
rds this being a 5th grade in the rather conservative for german standards bavaria i was
confused i looked down at the list again making sure i had read the name right look back
up at the kid me you are sandra kid yep me oh sorry thinking the kid must be from somewh
ere where sandra is both a girls and boys name where are you from i have only ever heard
that as a girls name before the class starts laughing sandra gets really quiet i am a gi
rl she says some of the other students start saying that their parents made the same mis
take when they met sandra i feel so sorry and stupid i get the class to calm down and fi
nish taking attendance we watch the movie in silence after the movie when we walked down
to where the poster presentation took place i apologised to sandra i felt so incredibly
terrible i still do to this day throughout the rest of the day i heard lots of whispers
about sandra i tried to stop them whenever they came up but there was no stopping the 5t
h grade gossip i had set in motion sandra if you are out there i am so incredibly sorry
for humiliating you in front of your class i hope you are happy and healthy and continue
to live your life the way you like do not let anyone tell you you have to dress or act a
certain way just because of the body parts you were born with i am sorry if i made you f
eel like you were wrong for dressing and acting differently i am sorry i probably made t
hat day hell for you i am sorry for my ignorance', 'tifu_by_gender_stereotyping', 0)
```

In [44]:

```
#set up corpus the lsh model will get files from

import sys
import os

print(os.path.abspath(os.getcwd()))
```

```
path_to_corpus = os.path.abspath(os.getcwd())+'\\corpus\\'

doc_num = 0

for doc in docs:
    name_of_file = '{}'.format(doc[1])
    completeName = os.path.join(path_to_corpus, name_of_file+".txt")

    with open(completeName, "w",encoding='utf-8') as f:
        f.write(doc[0])
    f.close()

    doc_num += 1
```

```
C:\Users\inbox\Desktop\CSC421\Project
```

In [93]:

```
#the LSH programn from [8]

import time, os
import shingling
import minhashing
import lsh
import statistics

def startLSH():
    print("\n*** Build the LSH buckets ***\n")

    # step 1: shingling
    timer_start = time.time()   # start timer
    folderpath = "corpus"       # path to corpus
    extension=".txt"            # specified extensions to read. Set to None to ignore e
    shingle_size = 10           # size of shingle: 8-12 is reommended
    shingle_matrix, files = shingling.get_shingle_matrix(folderpath, shingle_size, exte
    print(shingle_matrix.shape)
    print(f"Time taken for shingling: {time.time()-timer_start}")

    # step 2: min-hashing
    start_time = time.time()    # start timer
    no_of_hash_functions = 100   # specify no of hash functions for signature matrix
    incidence_matrix = read_pickle("corpus_inc_mat.pickle")
    signature_matrix = generate_signature_matrix(incidence_matrix, no_of_hash_functions
    print(f"Time taken for minhashing: {time.time()-start_time}")

    # step 3: LSH(Locality sensitive hashing)
    start_time = time.time()    # start timer
    r = 2
    buckets_list = lsh.get_bucket_list(signature_matrix, r)
    print(f"Time taken for lsh: {time.time()-start_time}")

    return buckets_list, signature_matrix, r, files


from pandas import read_pickle
import pandas as pd
import minhashing

def generate_signature_matrix(incidence_matrix, no_of_hash_functions):
    incidence_matrix = read_pickle("corpus_inc_mat.pickle")
```

```
    rows, cols = incidence_matrix.shape
    hashes = minhashing.generate_hash_functions(rows, no_of_hash_functions)
    signature_matrix = pd.DataFrame(index=[i for i in range(no_of_hash_functions)], col
    incidence_matrix = incidence_matrix.to_numpy()

    # core minhashing algorithm
    for i in (range(rows)):
        for j in range(cols):
            if incidence_matrix[i][j]==1:
                for k in range(no_of_hash_functions):
                    if np.isnan(signature_matrix[k][j]):
                        signature_matrix[k][j] = hashes[k](i)
                    else:
                        signature_matrix[k][j] = min(signature_matrix[k][j], hashes[k](

    print("saving generated signature_matrix to pickle file...")
    signature_matrix = pd.DataFrame(signature_matrix, columns = range(signature_matrix.
    signature_matrix.to_pickle("sig_mat.pickle")
    print("saved to sig_mat.pickle")
    return signature_matrix
```

In [ ]:

In [94]:
```
buckets_list, signature_matrix, r, documents = startLSH()
```

```
*** Build the LSH buckets ***

Using already created corpus_inc_mat.pickle file
using pickled file list
(127339, 97)
Time taken for shingling: 0.5810034275054932
saving generated signature_matrix to pickle file...
saved to sig_mat.pickle
Time taken for minhashing: 62.707231760025024
Time taken for lsh: 0.14058256149291992
```

In [95]:
```
print(documents)
```

```
[('corpus\\tifi_by_being_high_and_foreign.txt', 0), ('corpus\\tifu_after_a_party_and_wok
e_up_in_my_old_apartment.txt', 1), ('corpus\\tifu_and_seriously_over_did_it_in_zante.tx
t', 2), ('corpus\\tifu_a_lot.txt', 3), ('corpus\\tifu_by_accidentally_letting_my_0_year_
old_brother_see_my_sexts.txt', 4), ('corpus\\tifu_by_accidentally_sending_a_link_of_nude
_photos_to_a_friend.txt', 5), ('corpus\\tifu_by_accidentally_texting_the_most_attractive
_girl_at_my_school_prepare_your_anus_then_got_a_noise_disturbance_citation_from_the_poli
ce.txt', 6), ('corpus\\tifu_by_adding_commentary.txt', 7), ('corpus\\tifu_by_asking_for_
the_truth_from_someone_i_would_have_been_in_a_loving_relationship_with.txt', 8), ('corpu
s\\tifu_by_assuming_by_boyfriend_and_i_were_alone.txt', 9), ('corpus\\tifu_by_assuming_i
_was_home_alone.txt', 10), ('corpus\\tifu_by_being_a_part_of_a_convoy.txt', 11), ('corpu
s\\tifu_by_being_in_a_crowded_bus.txt', 12), ('corpus\\tifu_by_believing_my_last_final_p
aper_was_due_tomorrow.txt', 13), ('corpus\\tifu_by_breathing_in_dichloromethane_at_my_re
search_lab.txt', 14), ('corpus\\tifu_by_closing_out_the_bank_account_that_i_had_schedule
d_my_tax_returns_to_direct_deposit_into.txt', 15), ('corpus\\tifu_by_coughing_so_hard_i_
shat_my_self.txt', 16), ('corpus\\tifu_by_doing_a_zombie_impression_in_the_most_inapprop
riate_situation_imaginable.txt', 17), ('corpus\\tifu_by_doing_the_wrong_thing_in_the_rig
ht_place.txt', 18), ('corpus\\tifu_by_enjoying_a_few_minutes_outside_alone_at_0am_or_so_
```

i_thought.txt', 19), ('corpus\\tifu_by_failing_the_jersey_turnpike.txt', 20), ('corpus
\\tifu_by_falling_asleep_at_work_and_shitting_myself.txt', 21), ('corpus\\tifu_by_flicki
ng_my_bean_at_work.txt', 22), ('corpus\\tifu_by_flipping_my_boner_up_in_my_shorts_at_my_
girlfriends.txt', 23), ('corpus\\tifu_by_forcing_a_shart.txt', 24), ("corpus\\tifu_by_fu
cking_my_best_friend_'s_ex.txt", 25), ('corpus\\tifu_by_fucking_my_gf_in_the_woods.txt',
26), ('corpus\\tifu_by_gender_stereotyping.txt', 27), ('corpus\\tifu_by_getting_a_friend
_fired_from_work_at_least_it_was_due_to_her_own_actions_but_still.txt', 28), ('corpus\\t
ifu_by_getting_black_out_drunk_and_biting_my_wife.txt', 29), ('corpus\\tifu_by_getting_d
runk.txt', 30), ('corpus\\tifu_by_getting_in_a_car_accident.txt', 31), ('corpus\\tifu_by
_getting_rotten_robin_egg_juices_on_my_hand.txt', 32), ("corpus\\tifu_by_giving_my_profe
ssor_a_usb_drive_containing_'_horse_porn_'.txt", 33), ('corpus\\tifu_by_going_outside.tx
t', 34), ('corpus\\tifu_by_going_to_starbucks.txt', 35), ("corpus\\tifu_by_going_to_wend
y_'s.txt", 36), ('corpus\\tifu_by_having_a_spliff_in_the_bath.txt', 37), ('corpus\\tifu_
by_having_a_terrible_choice_in_mustache_at_the_wrong_time.txt', 38), ('corpus\\tifu_by_h
aving_sex_with_the_front_door_open.txt', 39), ('corpus\\tifu_by_hitting_on_my_burrito_gi
rl.txt', 40), ('corpus\\tifu_by_house_sitting_at_my_sisters.txt', 41), ('corpus\\tifu_by
_jumping_from_a_moving_vehicle.txt', 42), ('corpus\\tifu_by_keeping_my_stupid_mouth_shu
t.txt', 43), ('corpus\\tifu_by_laughing_at_the_wrong_time.txt', 44), ('corpus\\tifu_by_l
earning_that_my_local_police_force_do_indeed_do_their_jobs.txt', 45), ('corpus\\tifu_by_
listening_to_porn.txt', 46), ('corpus\\tifu_by_locking_myself_out_and_thinking_i_just_ha
d_gas.txt', 47), ('corpus\\tifu_by_losing_the_company_i_work_for_$_0k_usd.txt', 48), ('c
orpus\\tifu_by_making_a_horribly_inappropiate_joke.txt', 49), ('corpus\\tifu_by_making_s
oda.txt', 50), ("corpus\\tifu_by_mixing_taco_bell_and_arby_'s_in_one_day.txt", 51), ('co
rpus\\tifu_by_nofap_before_going_on_a_date.txt', 52), ('corpus\\tifu_by_not_checking_if_
the_bag_that_came_down_my_belt_actually_went_into_my_trailer_or_not.txt', 53), ('corpus
\\tifu_by_not_clipping_my_toenails.txt', 54), ('corpus\\tifu_by_not_having_proper_names_
for_the_numbers_in_my_phone.txt', 55), ('corpus\\tifu_by_not_knowing_the_difference_betw
een_needing_to_poop_and_needing_to_throw_up.txt', 56), ('corpus\\tifu_by_not_locking_the
_door_to_the_bathroom.txt', 57), ('corpus\\tifu_by_not_loosening_my_sneaker.txt', 58),
('corpus\\tifu_by_not_paying_attention.txt', 59), ('corpus\\tifu_by_not_submitting_my_co
llege_application_sooner.txt', 60), ('corpus\\tifu_by_not_using_three_bands.txt', 61),
('corpus\\tifu_by_overreacting_to_winning.txt', 62), ('corpus\\tifu_by_peeing_in_a_gator
ade_bottle.txt', 63), ('corpus\\tifu_by_petting_a_cow.txt', 64), ('corpus\\tifu_by_playi
ng_music_at_work.txt', 65), ('corpus\\tifu_by_pooping_in_the_woods_for_the_first_time.tx
t', 66), ('corpus\\tifu_by_prematurely_formatting_my_hard_drive.txt', 67), ('corpus\\tif
u_by_purchasing_a_large_double_ended_black_dildo.txt', 68), ("corpus\\tifu_by_putting_'_
headsooth_'_on_my_balls.txt", 69), ('corpus\\tifu_by_sharting_on_my_boyfriend.txt', 70),
('corpus\\tifu_by_showing_my_girlfriend_my_gay_porn.txt', 71), ("corpus\\tifu_by_signing
_my_crush_'s_yearbook_and_leaving_a_note_in_her_locker.txt", 72), ("corpus\\tifu_by_sitt
ing_in_my_friend_'s_lap.txt", 73), ('corpus\\tifu_by_sleeping_with_a_girl_other_than_the
_girl_i_like.txt', 74), ('corpus\\tifu_by_swallowing_a_half_dollar.txt', 75), ("corpus
\\tifu_by_tearing_through_a_friend_'s_fence_and_nearly_killing_myself.txt", 76), ('corpu
s\\tifu_by_telling_a_girl_i_hope_she_burns_herself.txt', 77), ('corpus\\tifu_by_telling_
my_dad_that_i_love_him.txt', 78), ('corpus\\tifu_by_testing_a_speaker.txt', 79), ('corpu
s\\tifu_by_thinking_i_had_to_urinate_and_in_turn_spraying_liquid_diarrhea_all_over_just_
in_time_for_an_elderly_man_to_walk_into_it.txt', 80), ('corpus\\tifu_by_throwing_away_0_
dollars.txt', 81), ('corpus\\tifu_by_trusting_a_fart_while_i_was_at_work.txt', 82), ('co
rpus\\tifu_by_trying_to_be_encouraging_to_a_teenage_girl.txt', 83), ('corpus\\tifu_by_tr
ying_to_defend_myself.txt', 84), ('corpus\\tifu_by_trying_to_help_an_elderly_man_use_ie_
on_windows_0.txt', 85), ('corpus\\tifu_by_trying_to_microwave_french_fries.txt', 86),
('corpus\\tifu_by_using_a_lock_on_my_locker_at_the_gym.txt', 87), ('corpus\\tifu_by_usin
g_public_transport.txt', 88), ('corpus\\tifu_by_wanting_a_pepperminty_bath.txt', 89),
("corpus\\tifu_by_watching_porn_at_my_grandmother_'s_house.txt", 90), ('corpus\\tifu_by_
writing_jokes_in_a_spiral.txt', 91), ('corpus\\tifu_during_sexy_times.txt', 92), ('corpu
s\\tifu_girl_gave_me_head_i_thought_i_camei_was_wrong.txt', 93), ('corpus\\tifu_trying_t
o_send_nude_pics_to_the_gf.txt', 94), ('corpus\\tifu_when_making_breakfast.txt', 95),
('corpus\\tify_by_experimenting_and_exploding_a_glass_bottle_inside_my_ass.txt', 96)]

Run the magic library

Given a file name return the file and all similar files as well as the file summaries

In [80]:
```
import glob
```

In [135…
```
sim_type = "jaccard"
while True:
    test_file = input("Enter file name or EXIT to quit: ")
    file_names = [os.path.basename(x) for x in glob.glob(os.path.abspath(os.getcwd())+'
    if test_file == "EXIT":
        break;
    if test_file in file_names:
        print(">> File does not exist in library.")
        continue
    for name,num in documents:
        if 'corpus\\'+ test_file == name:
            similar_docs = lsh.find_similar_docs(int(num), buckets_list, signature_matr
            break
    # test_file = int(input("Enter path of file: "))

    files = []

    for s in similar_docs:
        document = documents[s]
        document = document[0].rsplit('.', 1)
        for d in docs:
            if ('corpus\\'+d[1]) == document[0]:
                files.append(d)

    sumaries = []
    print('here are your similar files to {}'.format(test_file))
    for f in files:
        index = f[2]
        print(f[1],f[2])
        summary = build_text_summary(cleaned_data[index][0],cleaned_data[index][1],10,k
        sumaries.append((summary,cleaned_data[index][1]))

    print('summaries have been created. to view please exit')
print("\n*** Bye Bye :) ***\n")
print(sumaries)
```

```
Enter file name or EXIT to quit: tifu_by_putting_'_headsooth_'_on_my_balls.txt
here are your similar files to tifu_by_putting_'_headsooth_'_on_my_balls.txt
tifu_by_thinking_i_had_to_urinate_and_in_turn_spraying_liquid_diarrhea_all_over_just_in_
time_for_an_elderly_man_to_walk_into_it 95
tifu_by_putting_'_headsooth_'_on_my_balls 41
summaries have been created. to view please exit
Enter file name or EXIT to quit: EXIT

*** Bye Bye :) ***

[('no too many people and not an offense i need. i pull right up to the porta potty jump
out and leaving my car running bust inside. as i am taking an insanely high pressure pis
s this stream was truly powerful i feel itthatfeeling in the rear. i try to hold the rea
r ends contents in but the sheer force from my bladder is causing everything in that ent
ire region to let loose. after nodding my head and smiling i speed walk to my vehicle as
he opens the door to my mess. i could not bring myself to look in that direction and spe
d out of the parking lot as fast as i could.', ['tifu', 'by', 'thinking', 'i', 'had', 't
```

```
o', 'urinate', 'and', 'in', 'turn', 'spraying', 'liquid', 'diarrhea', 'all', 'over', 'ju
st', 'in', 'time', 'for', 'an', 'elderly', 'man', 'to', 'walk', 'into', 'it']), ('when m
y mate pipes up to my other mate hey man i dare you to put headsooth on your dick which
he understandably refused. he then throws it to me and as not one to back down from an a
greement i lathered that shit on. i run to the nearest sink and pull out my manhood and
what do i see. my balls shining like a beacon red as a london bus. the pain was unbearab
le i thought i was going to pass out. they are vigorously washing their balls and having
as much luck as i am.', ['tifu', 'by', 'putting', "'", 'headsooth', "'", 'on', 'my', 'ba
lls'])]
```

Bibliography

[1] https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25 accessed: April 2022 [2] https://www.analyticsvidhya.com/blog/2021/11/a-beginners-guide-to-understanding-text-summarization-with-nlp/ accessed: April 2022

[3] https://arxiv.org/abs/1706.03762 accessed: April 2022

[4] https://medium.com/@sylvanus.mahe/automatic-text-summarisation-c6b90482e414f accessed: April 2022

[5] https://www.sciencedirect.com/science/article/pii/S0306457320301047 accessed: April 2022

[6] https://github.com/wafaa-elkassas/EdgeSumm accesssed: April 2022

[7] https://klein.mit.edu/~hajiagha/NodePlanarSteiner.pdf accessed: April 2022

[8] https://github.com/RikilG/Locality-Sensitive-Hashing accessed: April 2022

[9] https://mrhasankthse.github.io/riz/2020/03/19/Minhash-and-LSH.html accessed: April 2022

[10] http://infolab.stanford.edu/~ullman/mmds/ch3.pdf accessed: April 2022

In [ ]:

In [ ]:

In [ ]:

In [ ]: