SYRACUSE UNIVERSITY
IST 664/CIS 668
Natural Language Processing(M003)
Spring 2021
Prof. Mei Zhang

FINAL REPORT

Group 7:
Vindhya Ravi Prakash - Computer Science
Aditya Tornekar - Applied Data Science
Rishabh Agarwal - Applied Data Science
Pawan S.P - Applied Data Science
Neil Malu - Computer Science

TABLE OF CONTENTS

1. INTRODUCTION

NLP is a branch of Artificial Intelligence that interacts between computers and humans using natural language. It basically helps computers to read, decipher and understand human languages. There are a lot of applications wherein NLP is beneficial. Some of the popular applications are Smart assistants, Language Translation, Text Generation, Text Summarization and many more. In this project, we are focusing on the Text Summarization.

Text Summarization is a technique of condensing a dense text into a concise version wherein it ensures that key terms and content are preserved. There are two types of Text Summarization.

1) **Extractive Text Summarization**

   It is a process of picking sentences from the document based on some scoring or other techniques to generate a summary. It recognizes important sections from a huge document and stitches them together to get a concise summary. This process is easier compared to the other summarizer and is popularly used everywhere.

2) **Abstractive Text Summarization**

   This summarizer generates completely new text by interpreting the original document and using advanced NLP techniques and Deep Learning concepts creates a new short summary. This is more powerful and gives better results when compared to Extractive Text Summarization. But, it is computationally too complex. Some of the trending models that use Abstractive techniques are Google's Transformer models: BERT, HuggingFace Libraries: GPT-2, T-5 and others.

2. MOTIVATION AND OUR GOAL

Text Summarization is widely used in a lot of areas and is very useful. Some of the popular applications are

1. Media Monitoring: To avoid content shock, text summarization can be used to condense contents.
2. SEO and Search Marketing: To be better than the competitors, companies need to analyse contents of other companies and figure out how their website comes at the top of the search list. To analyse a lot of websites is tedious, hence text summarization comes into play.
3. Google Search engine provides a short summary under every search result, highlighting the key terms.

But, our motivation stems from realizing the online nature of all events, lectures and meetings in the era of Zoom. Summarizing the Zoom audio transcripts for lectures, conferences and meetings can really help in a lot of ways. Sometimes conferences funded by government grants need the conference well documented and that requires generating summary for each session. Doing this manually is tedious. Hence, to save time and quickly generate summaries, we decided to use Text Summarization. This could be further extended by using it for classes/lectures to quickly get notes. Our goal for this project is to implement and analyze the various approaches of text summarization and compare them.

3. RELATED WORKS

3.1 EXTRACTIVE TEXT SUMMARIZATION

The following articles helped us with implementing extractive techniques of text summarization for our project:

1) **Text summarization using similarity matrix**

   The article compares and contrasts the two different types of text summarization techniques: Extractive and Abstractive summarization. Briefly describes abstractive text summarization and the emphasis is on extractive text summarization. It follows us through the process of extractive summarization using the similarity matrix which uses the cosine similarity to match the sentences. The flow of the code and also it discusses about the impacts of text summarization

2) **Text summarization using NLTK (weighted frequency)**

   This article uses the NLTK package for extractive text summarization of wikipedia articles by assigning the weighted frequency scores of the words to the sentences and picking the n top sentences with the highest sentence score. The python packages used in this article are NLTK, beautifulsoup, lxml, regex and heapq. The article first starts with scraping an article from wikipedia using the beautifulsoup and lxml package. Using regex, special characters and numbers are removed from the text and stored in a separate variable for computing the weighted frequency scores of the words. The main article text is then tokenized into sentences. The text without special characters and numbers are further preprocessed by removing the stopwords and tokenizing it into words. The word frequencies are computed for the words and the frequencies are then divided with the frequency of the most occurring word to compute the weighted frequency scores for the words. The weighted frequency scores are then allotted to the sentences. (Malik, 2021)Using the heapq library, the top 7 sentences with the highest scores are chosen as the summary.

3) **Different algorithms/approaches for extractive text summarization**

   This article gives an overview about the various approaches/algorithms which can be used for extractive summarization of text. The article talks about algorithms such as Latent Semantic Analysis (LSA), Luhn summarization, LexRank, KL-sum, etc. These various algorithms can be simply applied using the sumy library. Through this library itself, the text can be parsed and tokenized using the PlaintextParser and Tokenizer function in it. The text can then be summarized by passing the preprocessed text into the summarizer (Shrivarsheni, 2021).

3.2 ABSTRACTIVE TEXT SUMMARIZATION

Some of the related articles that inspired us or guided us with the implementation of Abstractive Text Summarization are:

1) **Sequence-to-Sequence Modeling with Attention Mechanism**

   This model uses LSTM i.e Long Short Term Memory along with Attention Mechanism to generate summary. Some of the packages that were used are numpy, pandas, regex, tensorflow (keras), NLTK tool kit and Attention model. The dataset used in this article is a News Summary and has two columns, headlines and text. They first clean the data by removing special characters like currencies, email, remove stopwords and perform contractions where short words are made long. For example, “ain’t” to “is not”. This is done to maintain uniformity in all of the texts. The dataset is then split into train and test where the Independent feature i.e X is the Text and the dependent feature i.e Y (need to be predicted) is the Headline. Tokenization is also performed and pad sequences are added to ensure that all the texts are of the same length. The model has an embedding layer, 3 LSTM layers for encoding and one LSTM layer for decode along with an attention mechanism. The dense layer also has an activation function i.e softmax. This model is trained by using the train set (where all the texts are first converted to sequence of vectors) and the result is the headline for all the text in the Test set.

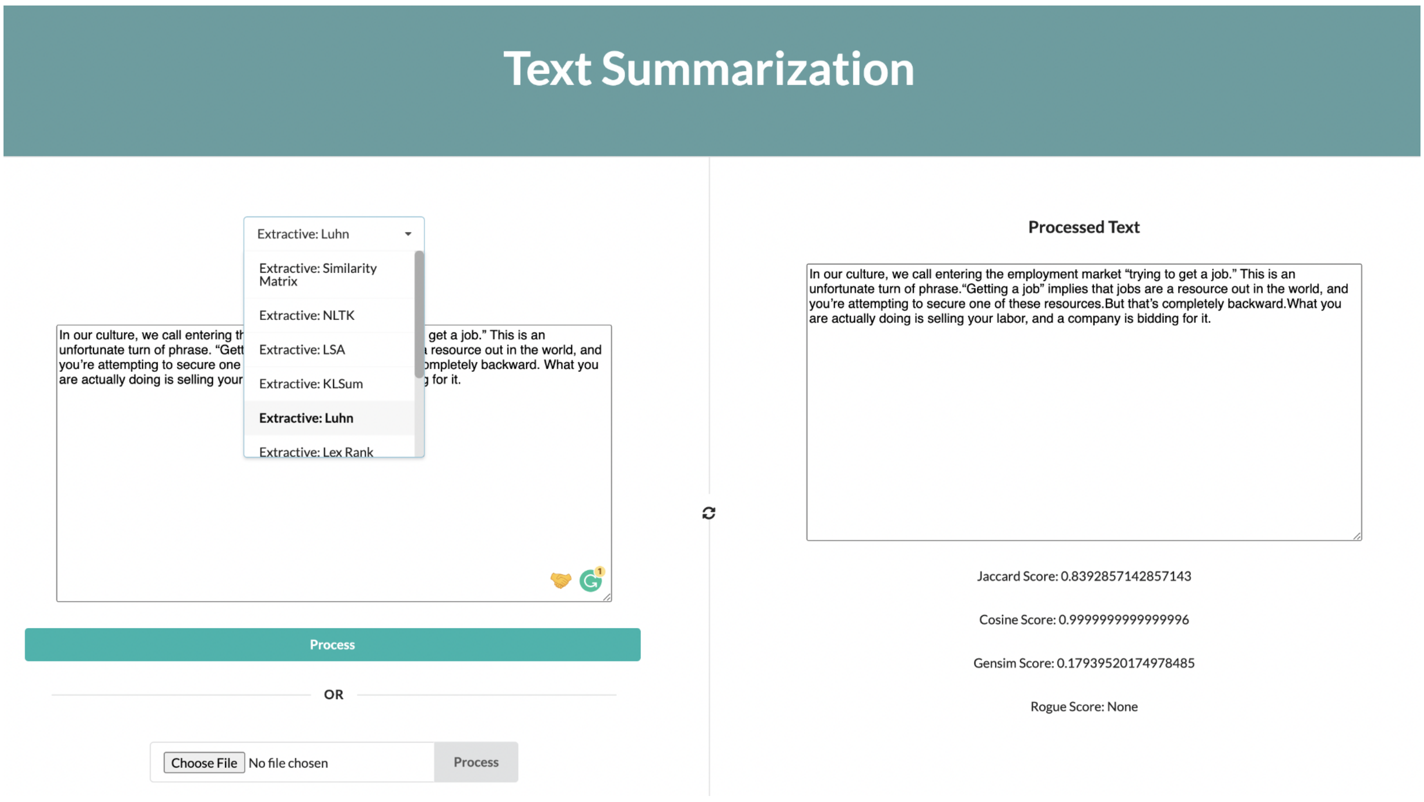
2) **Hugging Face Libraries**

   This article just gives an overview on the different Hugging Face models that are pretrained. Some of the models that were explained are T-5, BART, GPT-2 and XML. All these models are state-of-the-art transformer models and can be used for summarization, classification etc.

4. OUR APPROACH

4.1 FRAMEWORK

We have created a website using React and Django, where the user can input text or files ,and receive a summary with evaluation metrics based on the type selected. The files would be limited to be text based - eg. txt, pdf, html, docx, csv, vtt, jpeg.



In our culture, we call entering th | get a job." This is an
unfortunate turn of phrase. "Gett | resource out in the world, and
you're attempting to secure one | ompletely backward. What you
are actually doing is selling your | for it.

In our culture, we call entering the employment market "trying to get a job." This is an unfortunate turn of phrase."Getting a job" implies that jobs are a resource out in the world, and you're attempting to secure one of these resources.But that's completely backward.What you are actually doing is selling your labor, and a company is bidding for it.

Jaccard Score: 0.8392857142857143

Cosine Score: 0.9999999999999996

Gensim Score: 0.17939520174978485

Rogue Score: None

For reading the file, we are using the webvtt package which will extract the speaker name as well as the speech from the text and omit the timestamps.

The left subsection of the website is the user input, while the right section will display the summary as well as the evaluation metrics and scores described in Section 5.

4.2 DATA PREPROCESSING

We identify the main speaker based on who speaks the most, and remove noise (sentences that have less than 5 words). Sentences such as "Can you hear me" or "Can you see my screen" usually dominate these transcripts, and won't contribute to the summary.

4.3 EXTRACTIVE TEXT SUMMARIZATION

Extractive summarization uses the sentences from the original text as a summary for the text based on different methodologies for ranking a text. This helps in having a coherent summary as the sentences are from the original text itself.

4.31 EXTRACTIVE - SIMILARITY MATRIX

This method emphasizes the most significant parts of sentences and uses them to create the summary. The weights for the sentences are determined using various algorithms and methodologies, and the sentences are then ranked based on their importance and resemblance to one another. To detect sentence similarities and rank them, we use an unsupervised learning strategy. One advantage is that you won't have to train or develop a model before using it for our project. We created a similarity matrix which uses cosine similarity to check similarity between the sentences.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Since we are representing our sentences as a bunch of vectors, we use it to find the similarity among sentences. It measures the cosine of the angle between vectors. Angle will be **0** if sentences are perfectly similar.

The flow of the code looks like this :
Input document → sentences similarity → weight sentences → select sentences with higher rank.

4.32 EXTRACTIVE - NLTK (WEIGHTED FREQUENCY)

This method involves computing word frequencies for the text in the document and then computing weighted word frequencies by dividing the word frequencies by the frequency of the highest occurring word in the document. The weighted frequency scores are then allotted to the sentences to compute the sentence scores. The top N sentences are then chosen as the summary of the text (Malik, 2021).

The following steps were followed to implement this approach of extractive text summarization:

- Using regex, the stopwords, punctuation, numbers & special characters were removed from the main text & stored in a variable.
- Then the preprocessed text was tokenized at the word level and the word frequencies for the words was calculated. The weighted frequency for the words was computed by dividing the frequency by the frequency of the most occurring word.
- The unprocessed text was tokenized at the sentence level and the weighted frequencies for the words was assigned to the sentences to compute the sentence scores.
- The top N sentences with the highest score were chosen for the summary.

4.33 EXTRACTIVE - LSA

It is a form of unsupervised learning which extracts semantically significant sentences by applying singular value decomposition (SVD) to the matrix of term-document frequency.

4.34 EXTRACTIVE - KL Sum

It selects sentences based on similarity of word distribution as the original text. It aims to lower the KL-divergence criteria. It uses a greedy optimization approach and keeps adding sentences till the KL-divergence decreases (Shrivarsheni, 2021).

4.35 EXTRACTIVE - Luhn

The Luhn Summarization algorithm approach is based on TF-IDF (Term Frequency-Inverse Document Frequency). It is useful when very low frequent words as well as highly frequent words(stopwords) are both not significant. Based on this, sentence scoring is carried out and the high ranking sentences make it to the summary (Shrivarsheni, 2021).

4.36 EXTRACTIVE - Lex Rank

LexRank is an unsupervised approach to text summarization based on graph-based centrality scoring of sentences. The main idea is that sentences "recommend" other similar sentences to the reader. It is based on the idea that a sentence which is similar to other sentences in the text has a higher probability of being important (Shrivarsheni, 2021).

4.37 CHALLENGES FACED

Since in extractive summarization, sentences are extracted from the text and just concatenated to each other, there was no smooth transition between the ideas/concepts/topics in different sentences as the order of the sentences was changed. To circumvent this issue, we had to increase the number of top ranked sentences to be chosen for the summary.

4.4 ABSTRACTIVE TEXT SUMMARIZATION

Abstractive Text summarization extracts key information from the input text and then analyzes the sentiment and context to generate a sentence with similar meaning. So there are two main components to abstractive summarization: extraction and generation. The extraction of word features is a fairly simple process. However, generating text is a little trickier because the text generated needs to be similar to the input text, have analogous context, be sufficiently shorter than the input text and most importantly, **be coherent and readable by humans.** The last sub task is by far the toughest part of abstractive summarization.

There are a few ways we can go about generating input. In general, sequence modeling is a problem because documents are of variable length, so there needs to be some way of taking each document and meaningfully encoding it into a fixed size vector.

1. **Bag of words** - In Bag of words approach, each word in the vocabulary has one dimension. The English language has about 100,000 words in its vocabulary. So if you create a vector of 100,000 size, most of them would be '0' because most words wouldn't be in the document. But there is a key limitation because when you read a document, the order of the words matter, however in the BoW model, the order is irrelevant. Eg: the sequence "work hard" and "hard work" will have the same vectors and be considered equal. This is not good for a coherent summary. So the solution to that is N-grams, but the problem is the document size goes from 100,000 to $100{,}000^N$ which is computationally prohibitive.
2. **Recurrent Neural Network** - For documents with variable length, RNN's can be fed sequences to find the probability of the next potential word based on the sequences. However, since the document size is huge, there are a lot of layers in this network - so this leads to the problem of vanishing and exploding gradients.
3. **LSTM** - Stands for Long Short Term Memory. It is a kind of Recurrent Neural Network that uses better activation functions and has multiple gates which solve the problem of vanishing and exploding gradients. However, they still have some serious drawbacks - they are very difficult to train, and further, transfer learning never worked with LSTM, so it remained a supervised training algorithm because you needed a specific labelled dataset for each task - which just isn't appropriate for summarizing a user's input on the spot.
4. **Transformer** - They are the best known technique to summarize and generate text. And they work on unsupervised data. There are 3 kinds of transformer models - encoder based, decoder based and both.
    a. An encoder based model is good for representing text as a vector and therefore has a lot of contextual information. It also extracts key information such as frequency and semantics really well, and is therefore used in semantic analysis. This is extended to sentence classification. It is really good at summarizing text - so is great at Question/Answer applications, but is not good at generating coherent text. Eg. BERT - Bidirectional encoder ensures rich context information in both directions of each important word.
    b. A decoder based model is good at generating text because it calculates the probability distribution of the next word based on the previous sequence of

words. Eg - GPT-2 is trained on the web and generates pretty coherent sentences and can be fine tuned a lot but isn't great at summarizing.

c. Both - These are the best for abstractive text summarization since they do both tasks really well. Eg - HuggingFace's Pipeline API and Google's T-5 which is considered to be the gold standard.

4.41 ABSTRACTIVE

There are many algorithms that can be used for Abstractive Text Summarization. For each of the algorithms, similar data cleaning techniques were used, as for extractive summarization, so that the evaluation metrics were comparable at the end.

Some of the algorithms that we implemented are:

1. **Hugging Face Pipeline API**: This is an API that performs many operations like Named Entity Recognition, Sentiment Analysis, Feature Extraction, Text Summarization and many more. All these operations can be performed by calling the respective ops using the pipeline() function (encapsulation of all other pipelines).
2. **Google T-5 Model**: This is considered to be the best model in terms of both summarizing as well as generating text. It is also very customizable - however, because we could not figure out the best parameters based on the input, we did not get great results.
3. **GPT-2 Model**: This is an open source model that has been trained on the web so it has a vast knowledge base and contextual understanding. So it is really good at generating words fluently. It gave decent results, but also often just returned the original text.

4.43 CHALLENGES FACED

The biggest challenge we faced in abstractive text summarization was trying to implement LSTM with Sequence to Sequence and Attention Mechanism. Unfortunately this is supervised learning and it was really not feasible for us to create our own model and use it for different inputs of text.

Another challenge we faced was in implementing the Huggingface Pipeline API. The maximum character limit for this API is 1024 tokens, and any Zoom meeting summary, or even a webpage far exceeds this. There were 2 ways we could have gone about this - we could just break the text into 1024 tokens and then summarize each chunk and add them up. But this has a couple of significant drawbacks. First, this is very time consuming. Secondly, if there is a lot of noise in a single chunk, it's importance will increase and it will become part of the main summary.

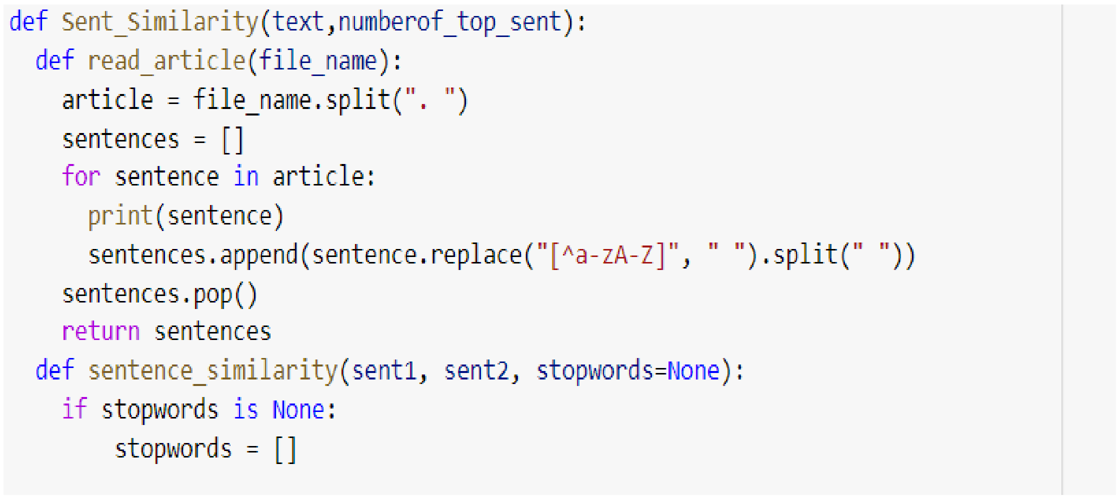
A better way of going about this was to first use the sentence similarity abstractive approach to get the top sentences, and then use the pipeline on these sentences only.

The last challenge we faced was tuning the hyperparameters for the T-5 and GPT-2 model. Unfortunately, these are completely dependent on the input and therefore we could not achieve great results.

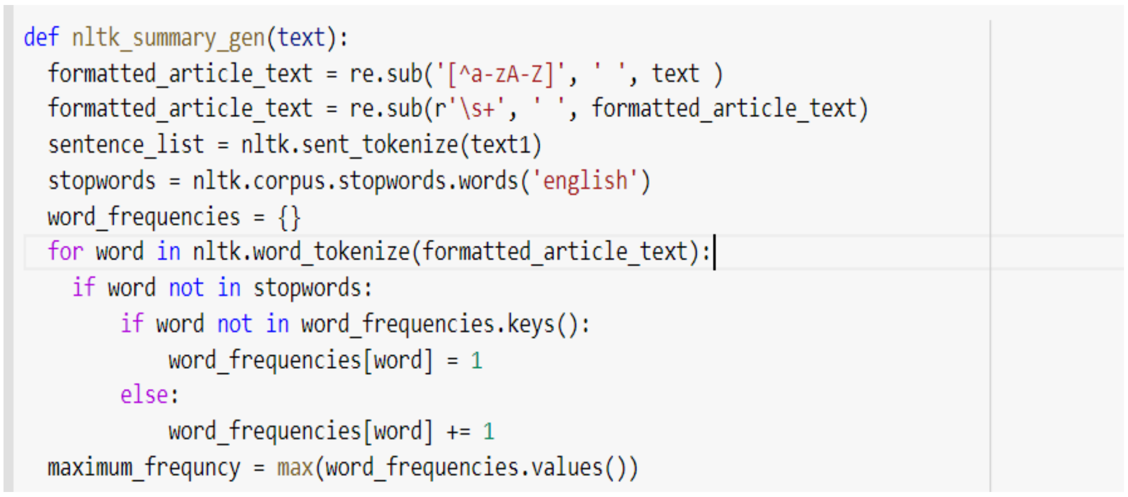
5. NLP TECHNIQUE USED

Some of the NLP Techniques used in this project are:

1) **Regex:** A sequence of characters i.e search pattern is used to perform string matching, finding or replacing some characters with another new character.
This is how we have used it in our project, mostly for cleaning purposes by replacing some pattern

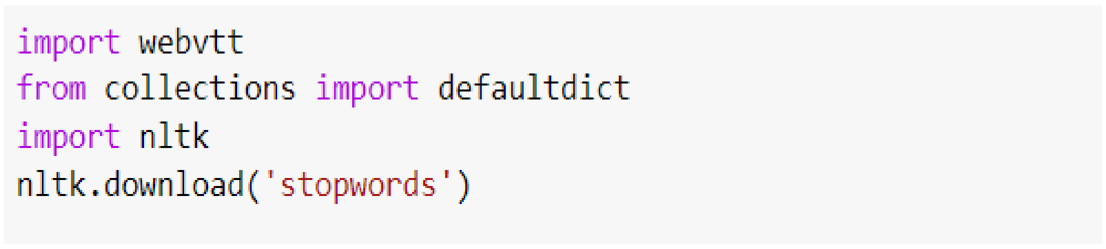


```
def Sent_Similarity(text,numberof_top_sent):
  def read_article(file_name):
    article = file_name.split(". ")
    sentences = []
    for sentence in article:
      print(sentence)
      sentences.append(sentence.replace("[^a-zA-Z]", " ").split(" "))
    sentences.pop()
    return sentences
  def sentence_similarity(sent1, sent2, stopwords=None):
    if stopwords is None:
        stopwords = []
```



```
def nltk_summary_gen(text):
  formatted_article_text = re.sub('[^a-zA-Z]', ' ', text )
  formatted_article_text = re.sub(r'\s+', ' ', formatted_article_text)
  sentence_list = nltk.sent_tokenize(text1)
  stopwords = nltk.corpus.stopwords.words('english')
  word_frequencies = {}
  for word in nltk.word_tokenize(formatted_article_text):
    if word not in stopwords:
        if word not in word_frequencies.keys():
            word_frequencies[word] = 1
        else:
            word_frequencies[word] += 1
  maximum_frequncy = max(word_frequencies.values())
```

2) **Stopword removal:** NLTK package provides a corpus of stop words that could be used to remove some words from our collection of words. The reason why we remove stop words is that these words have very little meaning like “is”, “an” and they unnecessary take up space. This helps in reducing the size of the list of words.



```
import webvtt
from collections import defaultdict
import nltk
nltk.download('stopwords')
```

```
stopwords = nltk.corpus.stopwords.words('english')
word_frequencies = {}
for word in nltk.word_tokenize(formatted_article_text):
  if word not in stopwords:
      if word not in word_frequencies.keys():
          word_frequencies[word] = 1
      else:
          word_frequencies[word] += 1
```

3) **Tokenization:** This is a process of reducing huge sentences into small tokens. This is mainly done to preprocess the data more. For example, pattern matching/replacing (using regex) or stemming and lemmatization. We have used the nltk package (in extractive) and keras package (in abstractive) to achieve this.

   Extractive:

```
for sent in sentence_list:
  for word in nltk.word_tokenize(sent.lower()):
      if word in word_frequencies.keys():
          if len(sent.split(' ')) < 30:
              if sent not in sentence_scores.keys():
                  sentence_scores[sent] = word_frequencies[word]
              else:
                  sentence_scores[sent] += word_frequencies[word]
```

4) **TF-IDF:** This is a statistical measure to find out how relevant a word is to the document. Mainly implemented in one of the Extractive Text Summarization techniques. There are two ways of implementing this, one is to find out the number of times the word occurs in the document and the other is to find the inverse document frequency of the word across a set of documents. This was mainly used for document search and information retrieval. Also, used in Gensim similarity evaluation approach for comparing the summary similarity using TF-IDF.

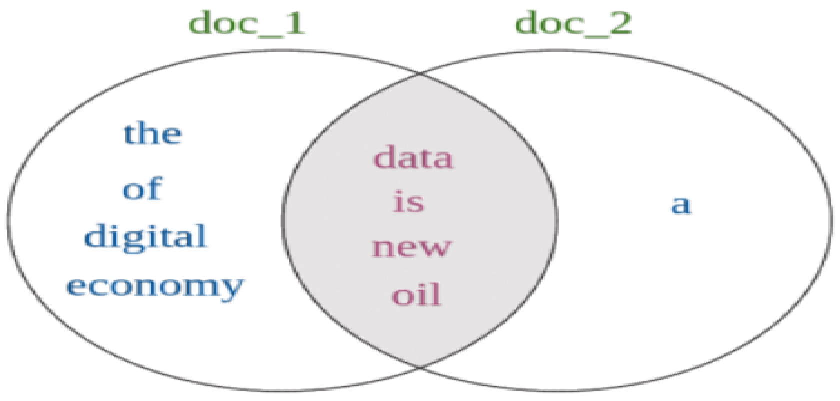
```
def get_gensim(generated_summary, actual_text_tosum):
  texts = actual_text_tosum
  keyword = generated_summary

  texts = [jieba.lcut(text) for text in texts]
  dictionary = corpora.Dictionary(texts)
  feature_cnt = len(dictionary.token2id)
  corpus = [dictionary.doc2bow(text) for text in texts]
  tfidf = models.TfidfModel(corpus)
  kw_vector = dictionary.doc2bow(jieba.lcut(keyword))
  index = similarities.SparseMatrixSimilarity(tfidf[corpus], num_features = feature_cnt)
  sim = index[tfidf[kw_vector]]
  x=[]
  for i in range(len(sim)):
    x.append(sim[i])
  return(sum(x)/len(x))
```

5) **Transformer models:** This is a neural network that effectively performs some of the nlp techniques. It converts one sequence to another with the help of two sections i.e Encoder and Decoder. Before RNN was popularly used. But, now BERT (by google) is popularly used to solve some of the issues in NLP.

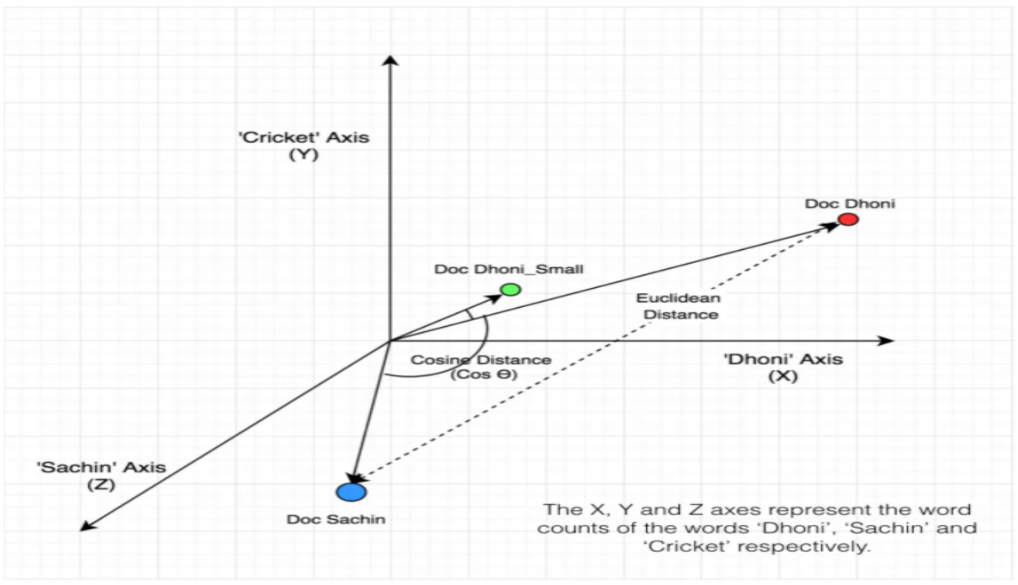
6) **Summary Evaluation:** For evaluation driven by usage of human-generated summary models, we have various methods such as ROGUE-scores, BLEU,NIST & METEOR. We thought we could use ROGUE-score using a summary generated by humans to evaluate and see which summary model performs better.
    a. ROGUE-Score: ROGUE-scoring mechanism uses human reference summary and generated summary to check the overlap between unigrams, bigrams from both summaries. It also checks for Longest Common Subsequence of n-grams and finally gives F1-Score, Precision and Recall values for all three categories. ROGUE-scoring is a highly accepted approach in the summarization domain.
    b. BLEU(not used in project as similar to ROGUE): BLEU is similar to ROGUE, but BLEU is more focused on precision mechanism, i.e. how many n-grams in the system generated summary are present in human reference summary, and ROGUE focusing on the converse of BLEU.

   These were some of the evaluation methods needing human intervention.

7) **Similarity Techniques:** One of the methods used for evaluation of the generated summary was using the similarity techniques. This evaluation method could be best for abstractive summarization because this summary is generated and we can check how well is the generated summary coinciding with the actual text, but is also a good indicator of evaluation for extractive summarization as well. We have used three techniques below:
    a. Jaccard Similarity: It is a technique which focuses on finding the intersecting words/tokens in the two texts from the union/overall text.



    b. Cosine Similarity: This technique uses TF-IDF vectors generated from the text and checks for the cosine angle between the vectors from generated summary and actual text. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. As the cosine value tends to approach 1, the smaller the angle between the two vectors and the greater is the match between vectors, in our case the two text documents.



c. Gensim Package: This is a python package useful to check generated similarity retrieval so that the summary generated can be evaluated.

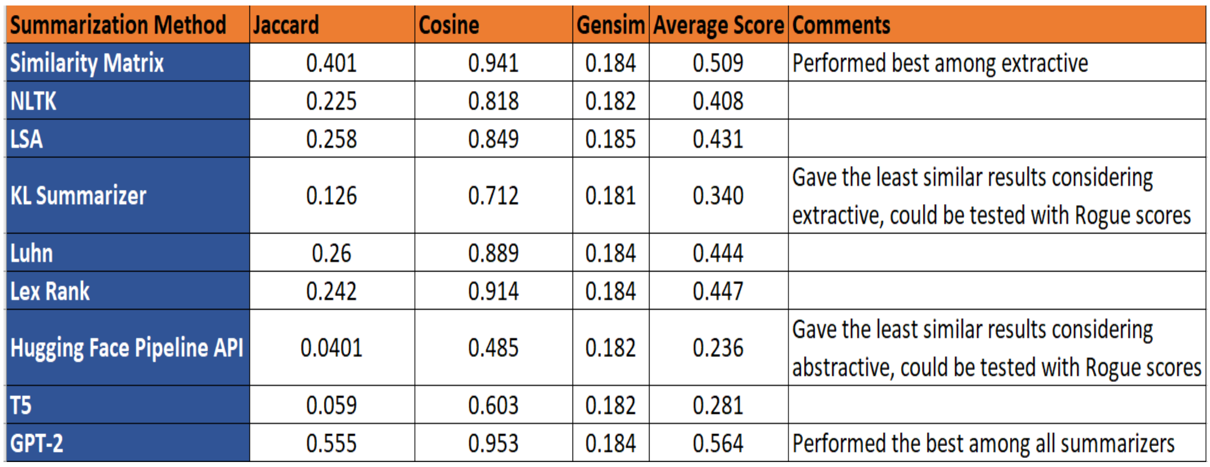
6. OBSERVATION

On implementation of the above summary scoring mechanisms using similarity techniques (Jaccard, Cosine and Gensim package), on the same actual text and summaries, we found that in general the Similarity Matrix approach showed good results in extractive methods. Even though the GPT-2 model was an abstractive approach, it showed overall best results in terms of summary similarity. KL Summarizer and Hugging Face Pipeline API showed the least similarity in the generated summary in extractive and abstractive approaches respectively. All these Summarization methods could be best evaluated with human-summaries using ROGUE-scores.

Although we have used the sentence similarity for evaluation of the summarizers, this only gives us a direction of which summarizer can possibly be our best candidate for summary generation and usage for a particular text. ROGUE-scores evaluation mechanism has been integrated into the code, but it requires human summary to run, which changes for different text files, and also the summary generated can be different for different humans passing their own summaries to evaluate.

This can be a possible reason for inconsistency or inaccuracy in the evaluation of the summarizers, where one human summary could be biased on a certain part of the text but the other human might not.

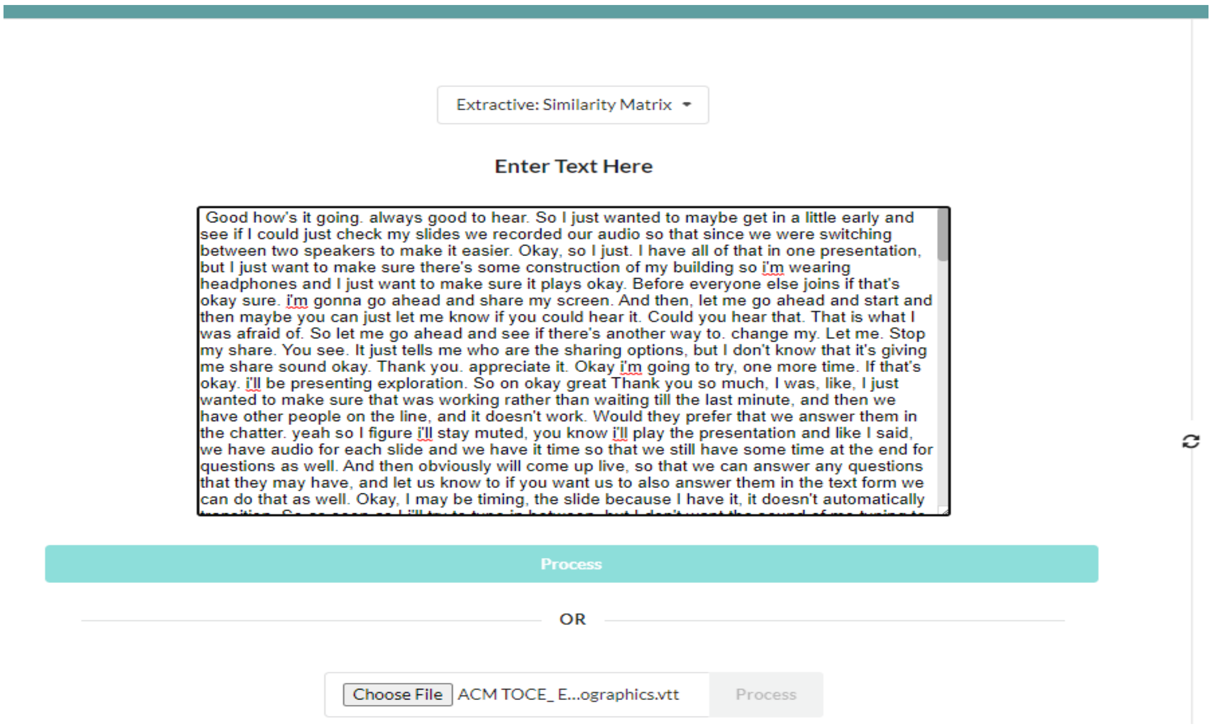
Also, the similarity evaluation is good for abstractive summaries as it checks the generated summary by the model using its own words to the actual summary. The extractive summary most likely will have good similarity as the sentences are extracted from the actual text itself, making it similar to the actual text.



Summarization Method	Jaccard	Cosine	Gensim	Average Score	Comments
Similarity Matrix	0.401	0.941	0.184	0.509	Performed best among extractive
NLTK	0.225	0.818	0.182	0.408	
LSA	0.258	0.849	0.185	0.431	
KL Summarizer	0.126	0.712	0.181	0.340	Gave the least similar results considering extractive, could be tested with Rogue scores
Luhn	0.26	0.889	0.184	0.444	
Lex Rank	0.242	0.914	0.184	0.447	
Hugging Face Pipeline API	0.0401	0.485	0.182	0.236	Gave the least similar results considering abstractive, could be tested with Rogue scores
T5	0.059	0.603	0.182	0.281	
GPT-2	0.555	0.953	0.184	0.564	Performed the best among all summarizers

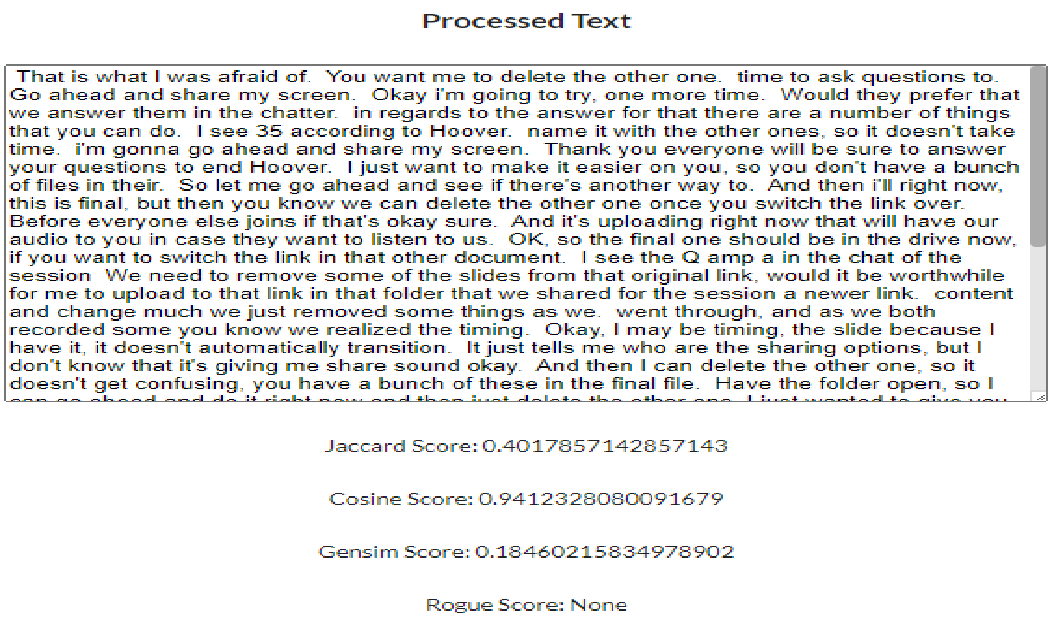
7. RESULTS

<u>Original Text:</u> Only the text needs to be entered or the file needs to be selected for getting the summary from the below text/vtt file.



Good how's it going. always good to hear. So I just wanted to maybe get in a little early and see if I could just check my slides we recorded our audio so that since we were switching between two speakers to make it easier. Okay, so I just. I have all of that in one presentation, but I just want to make sure there's some construction of my building so i'm wearing headphones and I just want to make sure it plays okay. Before everyone else joins if that's okay sure. i'm gonna go ahead and share my screen. And then, let me go ahead and start and then maybe you can just let me know if you could hear it. Could you hear that. That is what I was afraid of. So let me go ahead and see if there's another way to. change my. Let me. Stop my share. You see. It just tells me who are the sharing options, but I don't know that it's giving me share sound okay. Thank you. appreciate it. Okay i'm going to try, one more time. If that's okay. i'll be presenting exploration. So on okay great Thank you so much, I was, like, I just wanted to make sure that was working rather than waiting till the last minute, and then we have other people on the line, and it doesn't work. Would they prefer that we answer them in the chatter. yeah so I figure i'll stay muted, you know i'll play the presentation and like I said, we have audio for each slide and we have it time so that we still have some time at the end for questions as well. And then obviously will come up live, so that we can answer any questions that they may have, and let us know to if you want us to also answer them in the text form we can do that as well. Okay, I may be timing, the slide because I have it, it doesn't automatically

Extractive - Similarity Matrix



Processed Text

That is what I was afraid of. You want me to delete the other one. time to ask questions to. Go ahead and share my screen. Okay i'm going to try, one more time. Would they prefer that we answer them in the chatter. in regards to the answer for that there are a number of things that you can do. I see 35 according to Hoover. name it with the other ones, so it doesn't take time. i'm gonna go ahead and share my screen. Thank you everyone will be sure to answer your questions to end Hoover. I just want to make it easier on you, so you don't have a bunch of files in their. So let me go ahead and see if there's another way to. And then i'll right now, this is final, but then you know we can delete the other one once you switch the link over. Before everyone else joins if that's okay sure. And it's uploading right now that will have our audio to you in case they want to listen to us. OK, so the final one should be in the drive now, if you want to switch the link in that other document. I see the Q amp a in the chat of the session We need to remove some of the slides from that original link, would it be worthwhile for me to upload to that link in that folder that we shared for the session a newer link. content and change much we just removed some things as we. went through, and as we both recorded some you know we realized the timing. Okay, I may be timing, the slide because I have it, it doesn't automatically transition. It just tells me who are the sharing options, but I don't know that it's giving me share sound okay. And then I can delete the other one, so it doesn't get confusing, you have a bunch of these in the final file. Have the folder open, so I

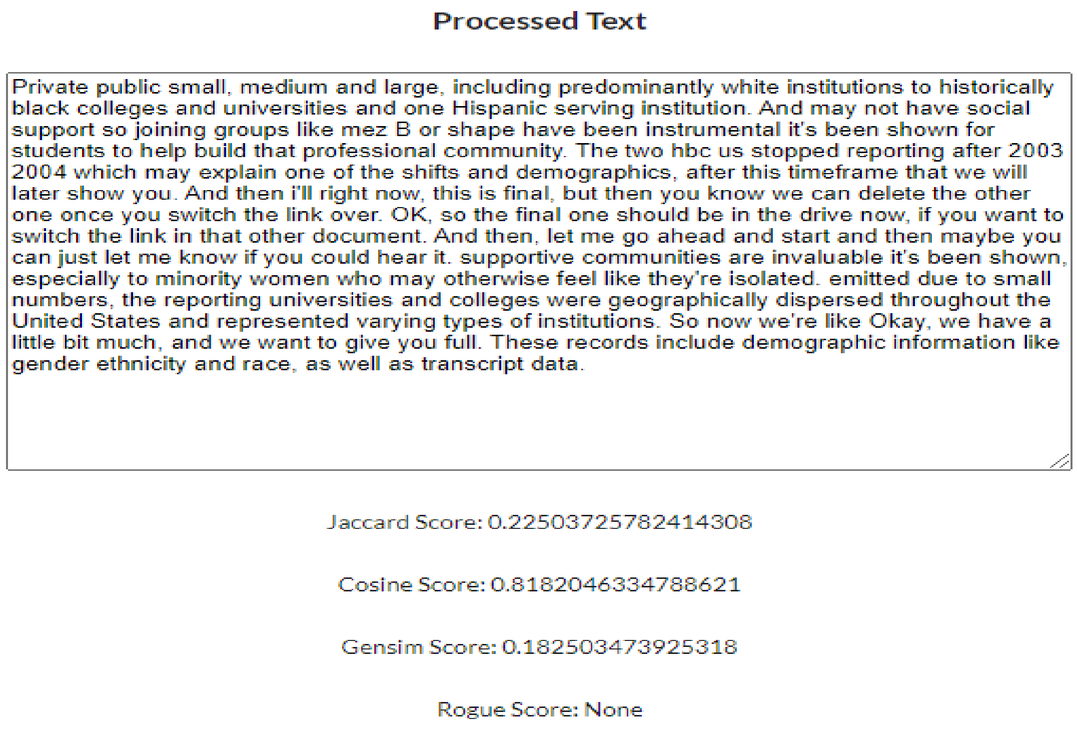
Jaccard Score: 0.4017857142857143

Cosine Score: 0.9412328080091679

Gensim Score: 0.18460215834978902

Rogue Score: None

Extractive - NLTK



Processed Text

Private public small, medium and large, including predominantly white institutions to historically black colleges and universities and one Hispanic serving institution. And may not have social support so joining groups like mez B or shape have been instrumental it's been shown for students to help build that professional community. The two hbc us stopped reporting after 2003 2004 which may explain one of the shifts and demographics, after this timeframe that we will later show you. And then i'll right now, this is final, but then you know we can delete the other one once you switch the link over. OK, so the final one should be in the drive now, if you want to switch the link in that other document. And then, let me go ahead and start and then maybe you can just let me know if you could hear it. supportive communities are invaluable it's been shown, especially to minority women who may otherwise feel like they're isolated. emitted due to small numbers, the reporting universities and colleges were geographically dispersed throughout the United States and represented varying types of institutions. So now we're like Okay, we have a little bit much, and we want to give you full. These records include demographic information like gender ethnicity and race, as well as transcript data.

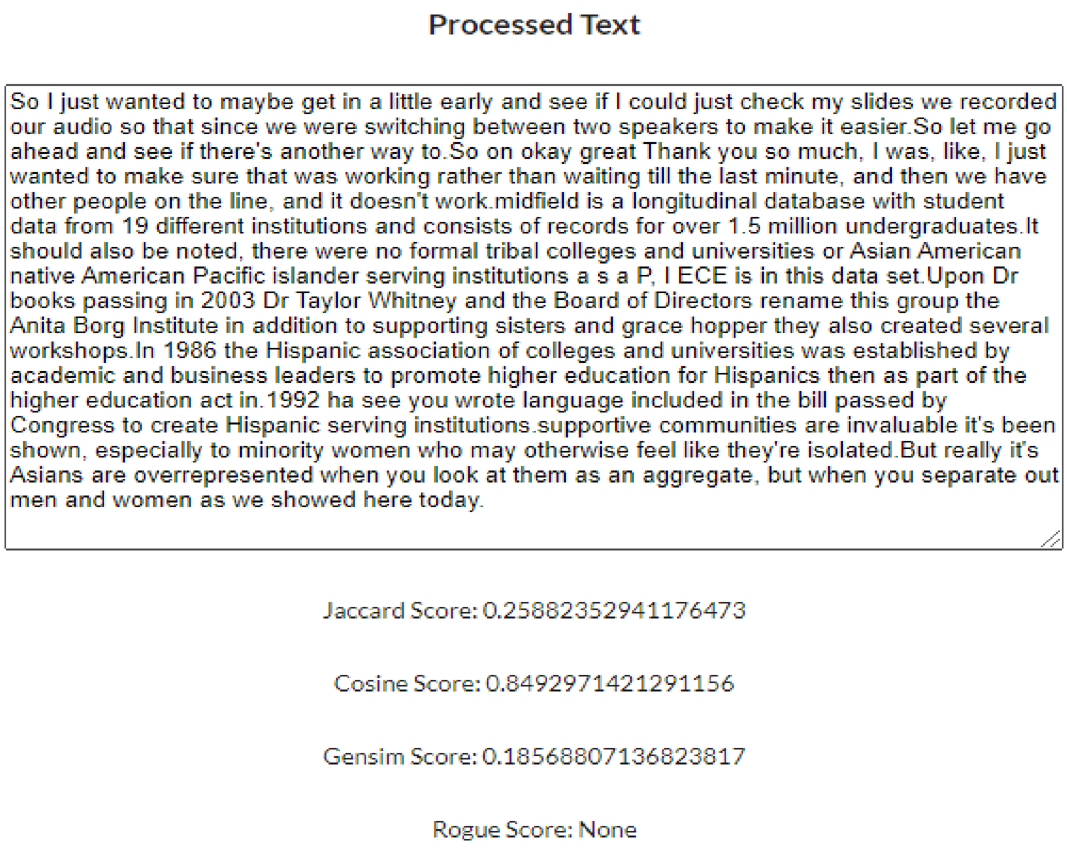
Jaccard Score: 0.22503725782414308

Cosine Score: 0.8182046334788621

Gensim Score: 0.182503473925318

Rogue Score: None

Extractive - LSA



Processed Text

So I just wanted to maybe get in a little early and see if I could just check my slides we recorded our audio so that since we were switching between two speakers to make it easier.So let me go ahead and see if there's another way to.So on okay great Thank you so much, I was, like, I just wanted to make sure that was working rather than waiting till the last minute, and then we have other people on the line, and it doesn't work.midfield is a longitudinal database with student data from 19 different institutions and consists of records for over 1.5 million undergraduates.It should also be noted, there were no formal tribal colleges and universities or Asian American native American Pacific islander serving institutions a s a P, I ECE is in this data set.Upon Dr books passing in 2003 Dr Taylor Whitney and the Board of Directors rename this group the Anita Borg Institute in addition to supporting sisters and grace hopper they also created several workshops.In 1986 the Hispanic association of colleges and universities was established by academic and business leaders to promote higher education for Hispanics then as part of the higher education act in.1992 ha see you wrote language included in the bill passed by Congress to create Hispanic serving institutions.supportive communities are invaluable it's been shown, especially to minority women who may otherwise feel like they're isolated.But really it's Asians are overrepresented when you look at them as an aggregate, but when you separate out men and women as we showed here today.

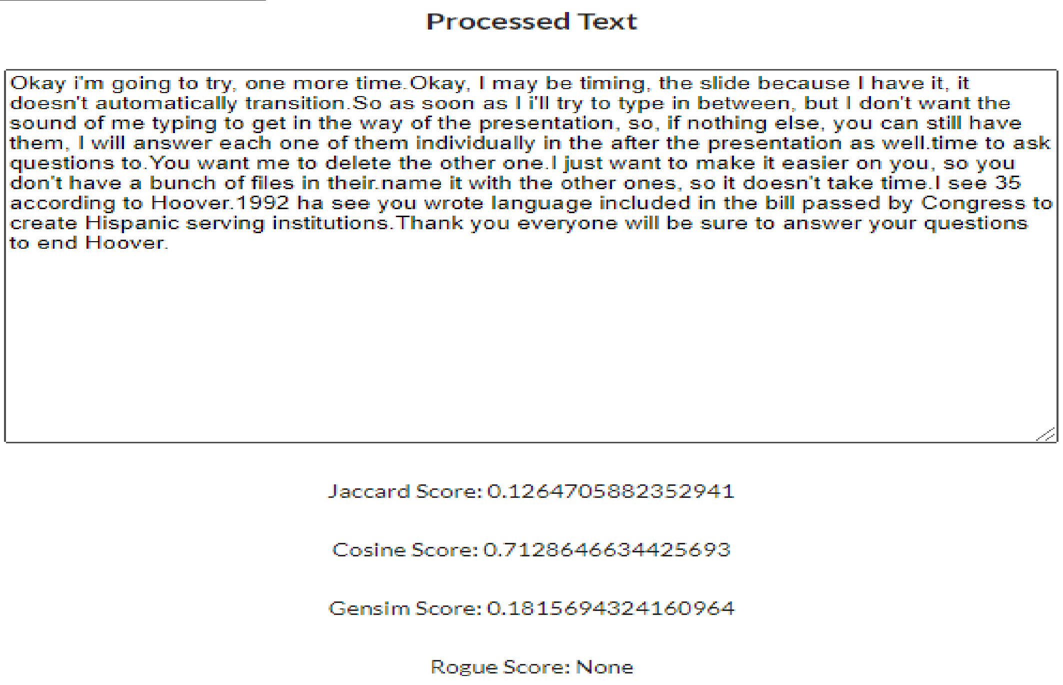
Jaccard Score: 0.25882352941176473

Cosine Score: 0.8492971421291156

Gensim Score: 0.18568807136823817

Rogue Score: None

Extractive - KL Sum



Processed Text

Okay i'm going to try, one more time.Okay, I may be timing, the slide because I have it, it doesn't automatically transition.So as soon as I i'll try to type in between, but I don't want the sound of me typing to get in the way of the presentation, so, if nothing else, you can still have them, I will answer each one of them individually in the after the presentation as well.time to ask questions to.You want me to delete the other one.I just want to make it easier on you, so you don't have a bunch of files in their.name it with the other ones, so it doesn't take time.I see 35 according to Hoover.1992 ha see you wrote language included in the bill passed by Congress to create Hispanic serving institutions.Thank you everyone will be sure to answer your questions to end Hoover.

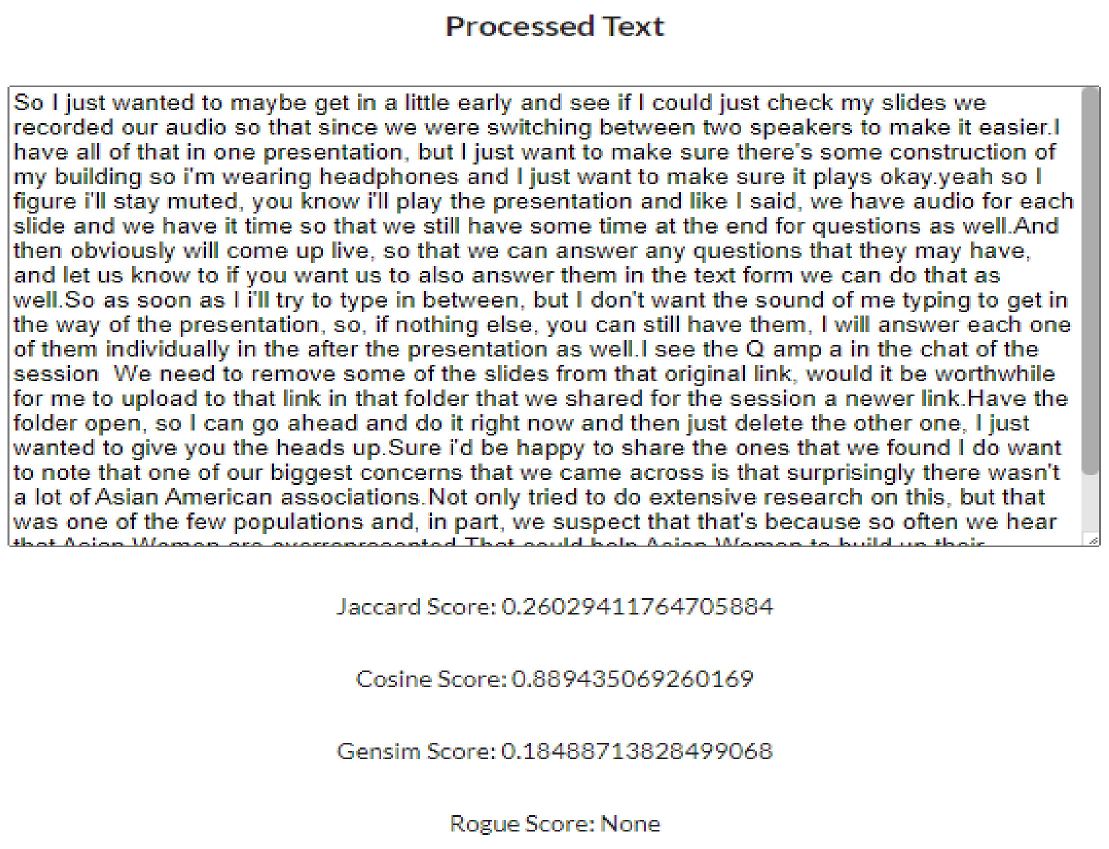
Jaccard Score: 0.1264705882352941

Cosine Score: 0.7128646634425693

Gensim Score: 0.1815694324160964

Rogue Score: None

Extractive - Luhn


Processed Text

So I just wanted to maybe get in a little early and see if I could just check my slides we recorded our audio so that since we were switching between two speakers to make it easier.I have all of that in one presentation, but I just want to make sure there's some construction of my building so i'm wearing headphones and I just want to make sure it plays okay.yeah so I figure i'll stay muted, you know i'll play the presentation and like I said, we have audio for each slide and we have it time so that we still have some time at the end for questions as well.And then obviously will come up live, so that we can answer any questions that they may have, and let us know to if you want us to also answer them in the text form we can do that as well.So as soon as I i'll try to type in between, but I don't want the sound of me typing to get in the way of the presentation, so, if nothing else, you can still have them, I will answer each one of them individually in the after the presentation as well.I see the Q amp a in the chat of the session We need to remove some of the slides from that original link, would it be worthwhile for me to upload to that link in that folder that we shared for the session a newer link.Have the folder open, so I can go ahead and do it right now and then just delete the other one, I just wanted to give you the heads up.Sure i'd be happy to share the ones that we found I do want to note that one of our biggest concerns that we came across is that surprisingly there wasn't a lot of Asian American associations.Not only tried to do extensive research on this, but that was one of the few populations and, in part, we suspect that that's because so often we hear that Asian Women are overrepresented.That could help Asian Women to build up their

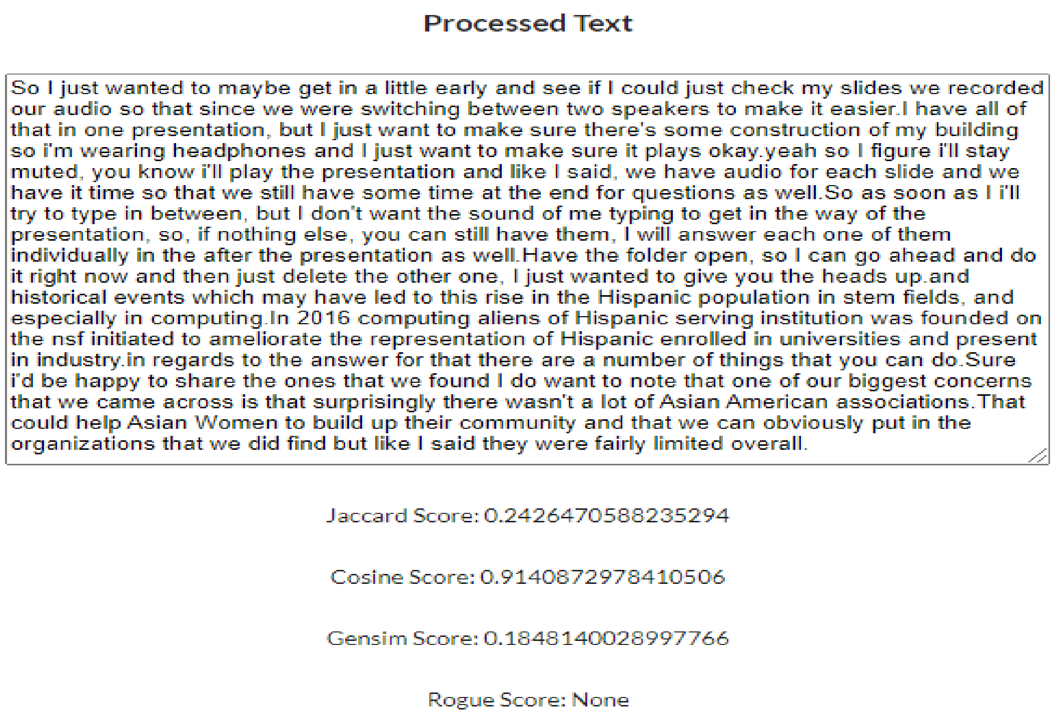
Jaccard Score: 0.26029411764705884

Cosine Score: 0.889435069260169

Gensim Score: 0.18488713828499068

Rogue Score: None

Extractive - Lex Rank


Processed Text

So I just wanted to maybe get in a little early and see if I could just check my slides we recorded our audio so that since we were switching between two speakers to make it easier.I have all of that in one presentation, but I just want to make sure there's some construction of my building so i'm wearing headphones and I just want to make sure it plays okay.yeah so I figure i'll stay muted, you know i'll play the presentation and like I said, we have audio for each slide and we have it time so that we still have some time at the end for questions as well.So as soon as I i'll try to type in between, but I don't want the sound of me typing to get in the way of the presentation, so, if nothing else, you can still have them, I will answer each one of them individually in the after the presentation as well.Have the folder open, so I can go ahead and do it right now and then just delete the other one, I just wanted to give you the heads up.and historical events which may have led to this rise in the Hispanic population in stem fields, and especially in computing.In 2016 computing aliens of Hispanic serving institution was founded on the nsf initiated to ameliorate the representation of Hispanic enrolled in universities and present in industry.in regards to the answer for that there are a number of things that you can do.Sure i'd be happy to share the ones that we found I do want to note that one of our biggest concerns that we came across is that surprisingly there wasn't a lot of Asian American associations.That could help Asian Women to build up their community and that we can obviously put in the organizations that we did find but like I said they were fairly limited overall.

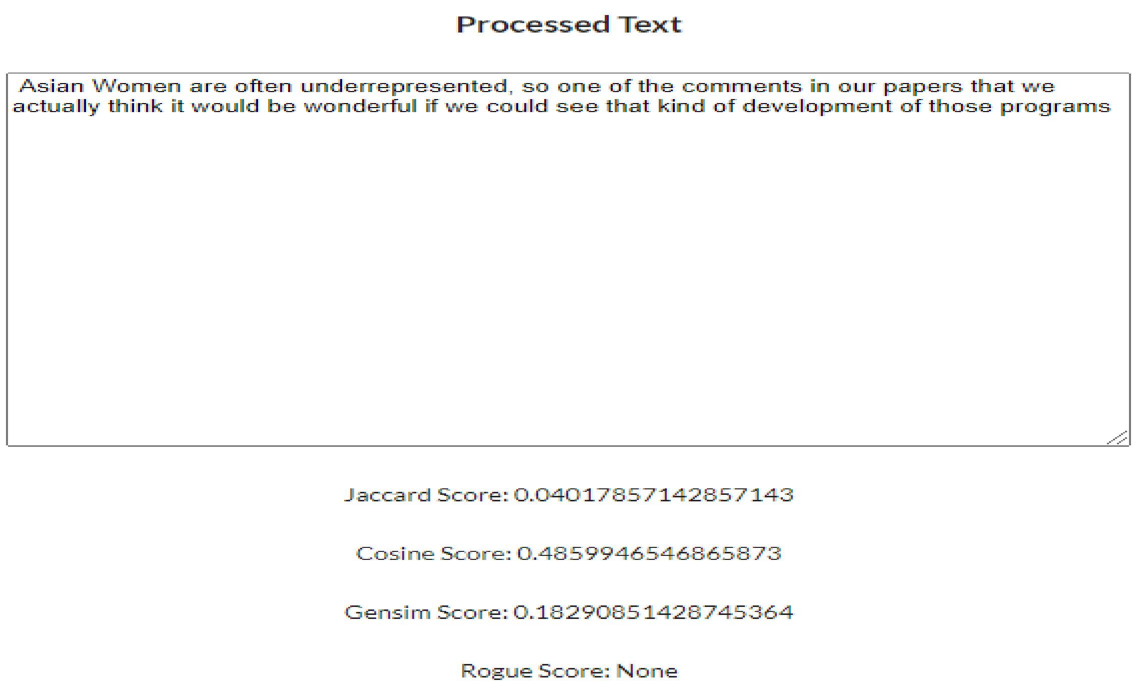
Jaccard Score: 0.2426470588235294

Cosine Score: 0.9140872978410506

Gensim Score: 0.1848140028997766

Rogue Score: None

Abstractive - Hugging Face Pipeline API



Processed Text

Asian Women are often underrepresented, so one of the comments in our papers that we actually think it would be wonderful if we could see that kind of development of those programs

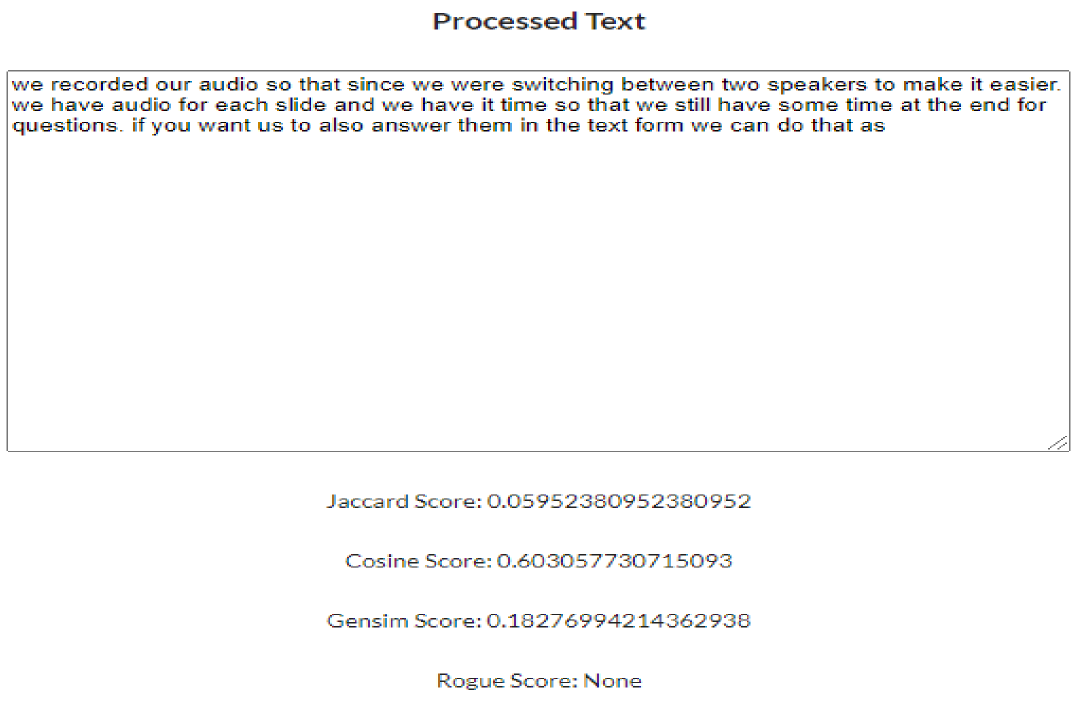
Jaccard Score: 0.04017857142857143

Cosine Score: 0.4859946546865873

Gensim Score: 0.18290851428745364

Rogue Score: None

Abstractive - T5



Processed Text

we recorded our audio so that since we were switching between two speakers to make it easier. we have audio for each slide and we have it time so that we still have some time at the end for questions. if you want us to also answer them in the text form we can do that as

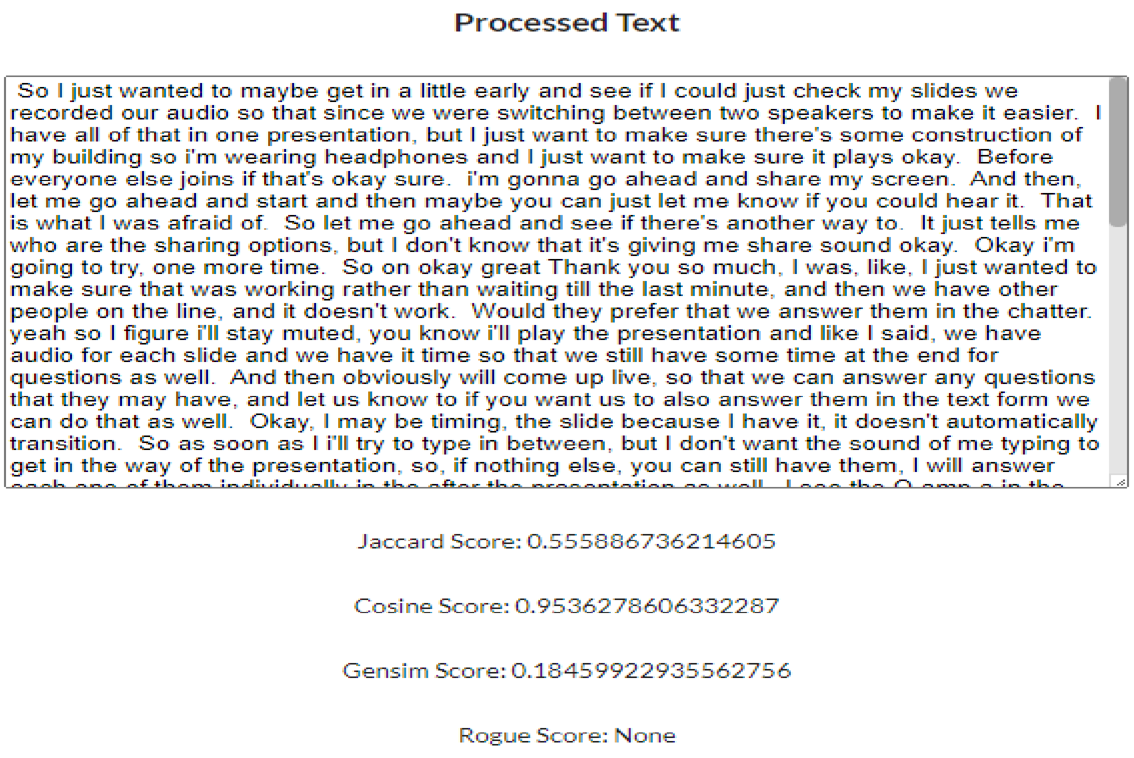
Jaccard Score: 0.05952380952380952

Cosine Score: 0.603057730715093

Gensim Score: 0.18276994214362938

Rogue Score: None

Abstractive - GPT-2



Processed Text

So I just wanted to maybe get in a little early and see if I could just check my slides we recorded our audio so that since we were switching between two speakers to make it easier. I have all of that in one presentation, but I just want to make sure there's some construction of my building so i'm wearing headphones and I just want to make sure it plays okay. Before everyone else joins if that's okay sure. i'm gonna go ahead and share my screen. And then, let me go ahead and start and then maybe you can just let me know if you could hear it. That is what I was afraid of. So let me go ahead and see if there's another way to. It just tells me who are the sharing options, but I don't know that it's giving me share sound okay. Okay i'm going to try, one more time. So on okay great Thank you so much, I was, like, I just wanted to make sure that was working rather than waiting till the last minute, and then we have other people on the line, and it doesn't work. Would they prefer that we answer them in the chatter. yeah so I figure i'll stay muted, you know i'll play the presentation and like I said, we have audio for each slide and we have it time so that we still have some time at the end for questions as well. And then obviously will come up live, so that we can answer any questions that they may have, and let us know to if you want us to also answer them in the text form we can do that as well. Okay, I may be timing, the slide because I have it, it doesn't automatically transition. So as soon as I i'll try to type in between, but I don't want the sound of me typing to get in the way of the presentation, so, if nothing else, you can still have them, I will answer

Jaccard Score: 0.555886736214605

Cosine Score: 0.9536278606332287

Gensim Score: 0.18459922935562756

Rogue Score: None

8. FUTURE WORK AND CONCLUSION

We used different summarization methods on the same Zoom vtt file and by looking at the results, we observed that GPT-2 from Abstractive and Extractive - Similarity Matrix approach gave good results. KL Summarizer from extractive and Hugging Face from Abstractive did not perform well for the Zoom vtt text file that we tested. Overall, all these summarizers could be further tested by passing human summary models for evaluation using ROGUE-Scores, which has been implemented in the module.

This could be further experimented in the future using below techniques:

- ROGUE - Score Evaluation using Human Summary Model
- BART Abstractive Approach
- Skip Thoughts Extractive Approach
- Seq2seq model with attention mechanism and implementation for large datasets
- Generate summary for Ted -Talks
- Generate summary from Meeting Minutes

9. DELIVERABLES

For this project we are submitting the following deliverables:

- **Website:** http://retr0-su-nlp.duckdns.org/
  This is the live website where the users can test the application and actively use it to summarize content, or analyze the different types of summarization, evaluate them and compare results.
- **GitHub:** https://github.com/neilmmalu/text-summarization
  This has all the source code used to create the website. There is a frontend, backend as well as some sample datasets that can be used to test it.
- **Python files:** We are also submitting two files - extractive.py and abstractive.py. These are the core summarization files used in the application - they are also available on the github repository. We wanted to highlight these files because they use the core NLP techniques - so we extracted them and added main methods so the user can summarize on the python REPL or terminal if they wish to.

10. REFERENCES

- Text Summarization Approaches for NLP – Practical Guide with Generative
- Text Summarization with NLTK in Python
- Hugging face libraries
- Seq2Seq: Abstractive Summarization Using LSTM and Attention Mechanism
- NLTK Documentation
- Jaccard Similarity for Evaluation
- Gensim Package for Evaluation
- Cosine Similarity
- Rogue - Scoring for Evaluation