

ITCS 4111/5111 Intro to NLP**Text Summarization - Final Project Report**

Shashikant Jaiswal

Tejaswini Naredla

Anusha Balaji

*under the guidance of Dr. Samira Shaikh***Introduction**

After working for eight hours, commuting for another hour, and spending another four for daily commitments and responsibilities, the last thing a person would want to do is browse through half-a-dozen websites just to get a glimpse of the news. At that point, it would be extremely beneficial for the person to have short and concise summaries of daily news. Our proposed program seeks to meet user needs by being a one-stop solution to users for a quick round-up of the day's events. By being able to process any given news article on the fly, regardless of the domain of the article or the length of it, and providing a quick summary of the article, we believe our program will be an effective alternative for users to easily and quickly catch up on news and hence, improve his/her reading experience.

Given summarizers have the potential to speed up and ease user reading experience, automatic text summarization has various applications ranging from newsletters to legal contract analysis to question answering, literature analysis and scientific research. Regardless of the diversity in the contexts a summarizer can be potentially used in, its goal remains the same: topic ensuring coverage and readability.

The following sections of this report provide more details on our proposed solution to build a robust and effective summarizer. Specifically, we discuss our research followed by tools, model, and methodologies and finally, our findings and conclusions from building our own text summarizer.

Background & Related Work

As a team, we have done extensive research on automatic text summarization to strategize how to build an effective summarizer. During the preliminary stages of our research, we used NLP Progress website as a starting point to look into current state-of-the-art approaches, which led us to some of the following papers and ideas [1].

Text summarization can be categorized according to input type, output type, and purpose. When it comes to input type, it can be based on either a single-document or multi-document. Single-document is when a summary needs to be generated only for one particular piece of literature and multi-document is when there exists several documents on the same topic and generating only one summary makes more sense for overlapping

and redundant corpora. As far as output type, text summarization can be classified as extractive or abstractive. While extraction-based summarization extracts words, phrases, or sentences directly from the corpus, abstractive summary seeks to generate summary from a semantic representation of the text.

Finally, text summarization model depends on purpose of the application. For instance, if the purpose is domain-specific, ex. celebrity gossip, multi-document summarization with feature based naive-bayes classifier works much better than single-document with deep learning. In contrast, if the summarizer is to be generic and handle any or all kinds of input, an ensemble model with multiple documents might work better. Similarly, the techniques to model text summarizer might also change if it is used more for query-based applications.

In “A survey of text summarization techniques”, the authors outlined numerous approaches for identifying important content for automatic text summarization. The authors Nenkova and McKeown particularly discuss various models ranging from the simple and naive frequency-based model, Bayesian model, to more complex ones including sentence clustering for topic representation, textrank, latent semantic analysis, and neural networks. This document mainly served as a guide by providing an overview of techniques we could potentially use for own solution rather than providing specific algorithms and details. Hence, we used it as a guide to direct us in researching for specific details in implementation and effectiveness of the techniques.

In “Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency (TF_IDF)”, the authors present more details on the TF-IDF approach including the two types of summarization (extraction and abstraction) and specific formulas to calculate various evaluation metrics like precision, recall, and f-measure. In this paper, the authors go on to describe their own implementation and effectiveness of extraction based summarization using TF-IDF technique.

Unlike the statistical technique of TF-IDF, textrank is more of a graph-based ranking model. In “TextRank: Bringing Order into Texts”, the authors present this unsupervised technique to extract important sentences in documents. The basic idea is to represent each sentence in the corpus as a vertex and have weighted edges to connect “similar” or overlapping sentences. Similarity measures include cosine similarity, string kernels. Tokens overlapping, longest common subsequence, etc. Modeling their TextRank after Google’s PageRank algorithm, the authors write, “the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes”. Some advantages of this approach is that it is unsupervised, meaning it does not require any training data, easily adaptable to new domains/genres of text, and produces a more cohesive summary by following linked vertices in the graph representation and producing similar/“connected” sentences as output.

Project Model & Approach

Dataset

We used a trimmed version of the CNN dataset (Google Drive link [here](#), original can be accessed [here](#)), which was already well preprocessed using Stanford CoreNLP open source package. Sentences from the original news articles are present in .SENT files while human written summaries are present in .SUMM files. Original CNN dataset comprises of about 180,000 articles. Since text summarization takes a long time to process and generate, we choose to work with the first 4,000 articles for the purposes of this project. We used a 80-20 split where 3,200 articles were used to train the TF-IDF vectorizer and the models followed by 800 articles to test the models.

As part of data preprocessing, we did a basic linguistic analysis where we did sentence and word-level tokenization and stop word elimination. This step was followed by feature extraction and model fitting processes described below.

Models

1. TF-IDF Model

Given the subjective nature of text summarization, as a team we decided to focus on implementing different models to get a better feel for the field. Firstly, we implement the frequently used statistical technique using term frequency-inverse document frequency (TF-IDF). The Term Frequency-Inverse Document Frequency (TF-IDF) method employs a numerical statistic to reflect how important a word is in a document in the corpus (Salton et al., 1988). This method is often used as a weighting factor in information retrieval and text mining. After data preprocessing and word tokenization, TF-IDF scores are calculated for each word by passing the training data through the TFIDF vectorizer from the scikit-learn package. Later, we compute a TF-IDF score for each sentence in a text document to reflect its importance. Thirdly, we select sentences and assemble the final summary by sorting the sentences according to their TF-IDF scores in descending order, and the highest ranked sentences are picked to be part of the generated summary.

The formula used to calculate TF-IDF score for each word in the training corpus is:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
 df_i = number of documents containing i
 N = total number of documents

2. TextRank Model

In addition to the statistical TF-IDF model, we also implement TextRank model for single document summarization and this is also an unsupervised approach. Data preprocessing was followed by building the graph with weighted edges using cosine

similarity measure. Each vertex in the graph represents a sentence in the article and each edge connecting the vertices in the graph represents the similarity between the corresponding pairs of sentences. After computing scores for each sentence in the graph, the top few sentences (about four) are extracted to be included in the final predicted summary.

TextRank algorithm follows the classic Google PageRank algorithm. The formula to calculate rank of a vertex (web page or sentence, in our case) is as follows:

$$\text{PageRank}(A) = (1-d) + d (\text{PageRank}(T_1)/C(T_1) + \dots + \text{PageRank}(T_n)/C(T_n))$$

Where,

- PR(A) is the page rank of page A
- PR(Ti) is the page rank of pages Ti which link to page A
- C(Ti) is the number of outbound links on page Ti and d is the damping factor

***The damping factor d represents the probability of a user jumping to a web page at random as the next destination, ignoring the page link structure completely. Usually, the standard value for damping factor is 0.85 (see reference 8).

The following table summarizes our models, steps involved in building, and what tools and packages we used in implementing them:

Model	Tools & Packages
TF-IDF	<ul style="list-style-type: none"> ● NLTK for word and sentence tokenization ● Used scikit-learn Tf-Idf vectorizer to compute Tf-Idf vectors ● Fit model on train dataset ● Using the trained tf-idf model we generated summary for each test news article (documents with .sent extension) in the test corpus. ● Transform each document into tf-idf vector ● Sentence score = sum of tf-idf values of each word ● Used dictionary to store sentence and its score as key, value pair ● Sorted the items in dictionary based on values (sentence scores) in descending order ● Extracted top n (mostly 3) sentences and generated it as summary
TextRank	<ul style="list-style-type: none"> ● Similar to Google's PageRank algorithm where each page is represented as vertex and the edge represents the links between pages. And page rank value is the probability that user visits the page. ● Each sentence in .SENT file is a vertex and edges in the graph are computed using the TextRank algorithm. ● Scores of each vertex are updated iteratively as new edges are added or as edges are updated. ● Top n (mostly 3) vertices are extracted and the sentences become part of the extractive summary.

The following table displays what role each team member played in building our model and achieving our goals:

Team Member	Contributions
Tejaswini	Researched and implemented TextRank algorithm; provided support and helped debug problems in code implementation and execution
Shashikant	Implemented Rouge score evaluation; researched TF-IDF and centroid-based clustering models; documentation and reporting
Anusha	Helped implement TF-IDF model; researched feature extraction and centroid-based summarization; researched automatic text summarization techniques; documentation and deliverables

Evaluation

Given such subjectivity in text summarization, it is critical to devise a solid system to evaluate our model and decipher its usefulness when it comes to text summarization part of evaluation.

The provided summaries in the CNN dataset were human-written. Although it was mentioned the writers came up with extractive summaries, the sentence summaries were not exactly taken word-for-word from the articles. As it would be meaningless to count the number of overlapping sentences between our summary and the actual word-level extractive summary, we decided to instead use a popular metric called the Rouge score to evaluate our summaries.

Rouge score is a very popular and classic natural language evaluation metric used in the automatic text summarization domain. In our project, we used a readily available implementation posted on github [here](#). According to the github readme, the program takes in a parameter called alpha that can be adjusted to favor either recall (when alpha is close to 0) or precision (when alpha is close to 1) in our model. Recall quantifies how well the actual summary captures information contained in the machine generated summary, while precision correlates with how succinct is the machine generated summary compared to the actual summary.

Rouge score is computed using the below formula:

$$n = (1.0 - \alpha) * \text{precision_score} + \alpha * \text{recall_score}$$

$$\text{rouge_score} = (\text{precision_score} * \text{recall_score}) / n$$

***For our implementation, we rely on **Rouge-N (for N=1)** metric, which computes rouge score on unigrams in reference and produced summaries. This metric compares the number of tokens matching in both the summaries. Our results are presented in the next subsection.

Results:

a. Summarization using TF-IDF approach

Following is a few screenshots of our TF-IDF summarizer:

Screenshot 1:

The Reference summary:

Author Terry Pratchett has died age 66, his website says.
"In over 70 books, Terry enriched the planet like few before him," says publisher.

The model generated summary:

He said then, "Frankly, I would prefer it if people kept things cheerful, because I think there's time for at least a few more books yet."
"Shouting from the rooftops about the absurdity of how little funding dementia research receives, and fighting for good quality dementia care, he was and will remain the truest of champions for people with the condition."
According to Thursday's statement, he had posterior cortical atrophy, a progressive degenerative condition involving the loss and dysfunction of brain cells, particularly at the back of the brain.

Screenshot 2:

The Reference summary:

Police search co-pilot Andreas Lubitz's apartment for clues.
A pilot who knew Lubitz calls him a "very normal young person".
Investigators say they believe he deliberately crashed Germanwings Flight 9525.

The model generated summary:

At a club on the outskirts of Montabaur, pilots who knew Lubitz said they were shocked to hear what investigators said.
"(He was) a very normal young person, full of energy," Klaus Radke said.
"He was a lot of fun, even though he was perhaps sometimes a bit quiet," Ruecker said.

Screenshot 3:

The Reference summary:

Thousands march in a protest against terrorism in Tunisia's capital.
Demonstrators hold signs that say "We are not afraid" and "Je suis Bardo".
Tunisia's Prime Minister says a suspect in the Bardo museum attack was killed in a raid.

The model generated summary:

Protesters held banners that said "We are not afraid" and "Je suis Bardo" as they chanted "Tunisia is free, and out with terrorism."
On March 18, the art, culture and history museum was the site of a drastically different scene, as gunmen opened fire on tourists in a siege that also forced the evacuation of the neighboring Parliament.
"We came to express our support and to fight this danger that's threatening our society and our stability," said Rafik Abdessalem, Tunisia's former foreign minister, who was among the crowd.

Screenshot 4:

The Reference summary:

Tennis legend Roger Federer exchanges hairstyle tips with Andy Murray.
The men discuss curls, smoothness and "being worth it" over Twitter.
The sport is no stranger to funky hairstyles over the years.

The model generated summary:

Federer crashed the Scot's Q&A Twitter session on Thursday to tease Murray about his lustrous locks, asking "how do you get your hair to be so curly?"
Think it would be good for your image."
Earlier this month, Federer suffered some embarrassment of his own when a small boy played a point against the 33-year-old -- outmaneuvering the Swiss legend with a perfect lob at an exhibition match in New York.

Screenshot 5:

The Reference summary:

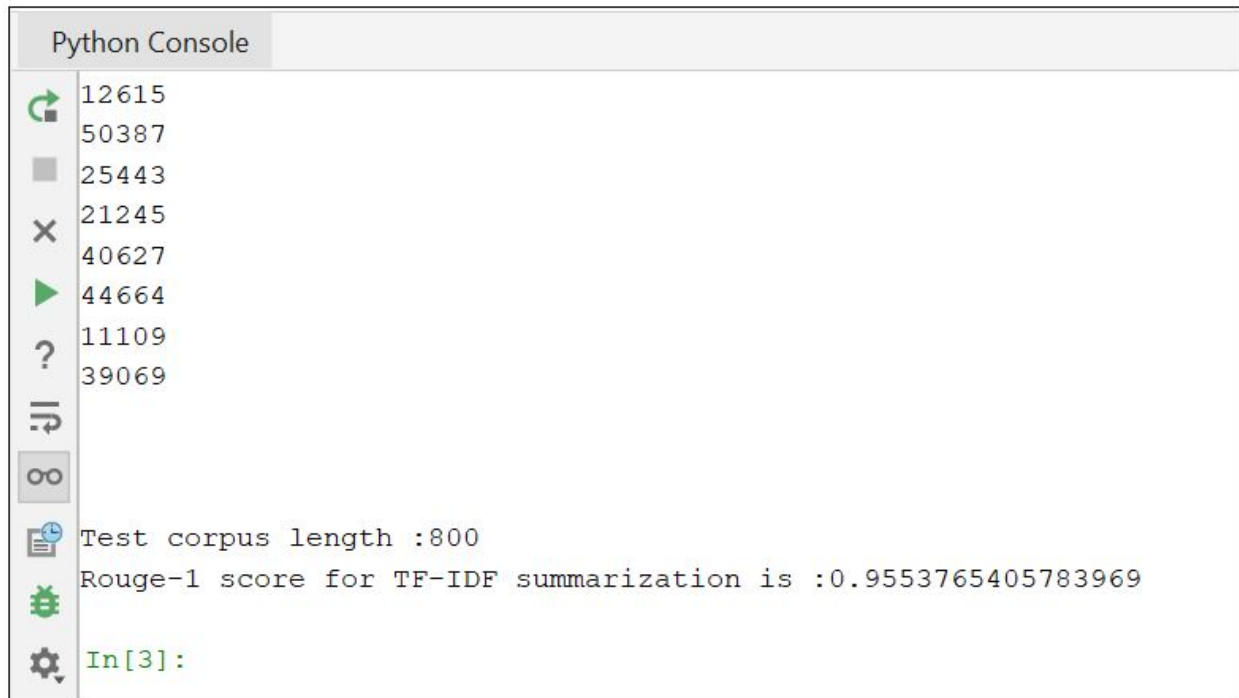
Amran Abdundi wins the campaigning category at 2015 Index Freedom of Expression awards.
Honored for her work helping victims of rape and conflict along Kenya-Somalia border.

The model generated summary:

CNN: How did you get involved in women's rights along the Kenyan-Somali border?
When I first began, I was just doing it in our village and now I move around all over the place up to the Somali border.
CNN: And how many women are crossing the border?

Rouge Score for TF-IDF Model

Rouge-1:



The screenshot shows a Jupyter Notebook interface with a 'Python Console' tab. On the left, there is a vertical toolbar with icons for running, saving, and other actions. The console output displays a list of numbers: 12615, 50387, 25443, 21245, 40627, 44664, 11109, and 39069. Below these numbers, the text 'Test corpus length :800' is shown. The final line of output is 'Rouge-1 score for TF-IDF summarization is :0.9553765405783969'. The prompt 'In[3]:' is visible at the bottom.

```

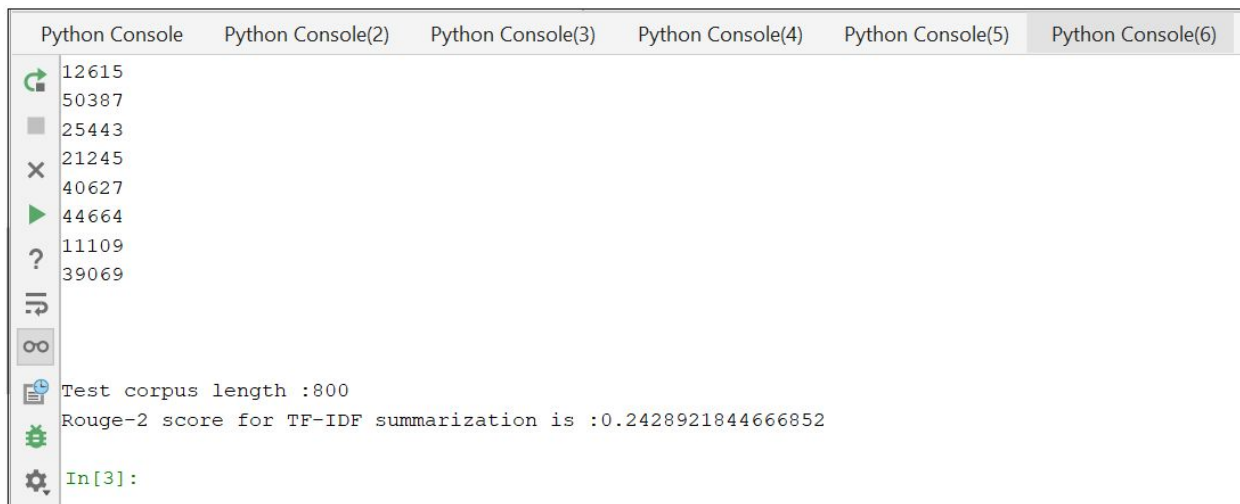
12615
50387
25443
21245
40627
44664
11109
39069

Test corpus length :800
Rouge-1 score for TF-IDF summarization is :0.9553765405783969

In[3]:

```

Rouge-2:



The screenshot shows a Jupyter Notebook interface with multiple 'Python Console' tabs. The active tab is 'Python Console(6)'. The console output displays the same list of numbers as the previous screenshot: 12615, 50387, 25443, 21245, 40627, 44664, 11109, and 39069. Below these numbers, the text 'Test corpus length :800' is shown. The final line of output is 'Rouge-2 score for TF-IDF summarization is :0.2428921844666852'. The prompt 'In[3]:' is visible at the bottom.

```

12615
50387
25443
21245
40627
44664
11109
39069

Test corpus length :800
Rouge-2 score for TF-IDF summarization is :0.2428921844666852

In[3]:

```

b. Summarization using TextRank

Following is a few screenshots of our TextRank summarizer:

Screenshot 1:

The Reference summary:

Author Terry Pratchett has died age 66, his website says.
"In over 70 books, Terry enriched the planet like few before him," says publisher.

The model generated summary:

Pratchett, who wrote more than 70 books, including those in his "Discworld" series, had been diagnosed with a rare form of Alzheimer's disease in 2007. The last posts on his verified Twitter account, run by Pratchett with close friend Rob Wilkins, give a poignant farewell -- and have already been retweeted thousands of times. "Shouting from the rooftops about the absurdity of how little funding dementia research receives, and fighting for good quality dementia care, he was and will remain the truest of champions for people with the condition."

Screenshot 2:

The Reference summary:

Police search co-pilot Andreas Lubitz's apartment for clues.
A pilot who knew Lubitz calls him a "very normal young person".
Investigators say they believe he deliberately crashed Germanwings Flight 9525.

The model generated summary:

Officials say Lubitz passed a psychological test when he was hired, had no known ties to terrorism and showed no sign of medical distress during the flight. Police spokesman Markus Nieszczy said a team of five investigators went "through the apartment looking for clues as to what the co-pilot's motivation might have been, if he did indeed bring the plane down." If it was for medical reasons, he said, then that information would have been private before the crash, he said, but it will be part of information gathered during the investigation.

Screenshot 3:

The Reference summary:

Thousands march in a protest against terrorism in Tunisia's capital.
Demonstrators hold signs that say "We are not afraid" and "Je suis Bardo".
Tunisia's Prime Minister says a suspect in the Bardo museum attack was killed in a raid.

The model generated summary:

As a heavy police presence stood guard, Tunisian President Beji Caid Essebsi marched alongside dignitaries and world leaders, including French President Francois Hollande and Italian Prime Minister Matteo Renzi, who led the crowd to the steps of the Bardo Museum. On March 18, the art, culture and history museum was the site of a drastically different scene, as gunmen opened fire on tourists in a siege that also forced the evacuation of the neighboring Parliament. Hours before Sunday's demonstration began, Tunisia's Prime Minister announced that Algerian national Khaled Shayeb, the alleged architect of the museum assault, was one of nine suspected militants killed in a raid in the south of the country.

Screenshot 4:

The Reference summary:

Tennis legend Roger Federer exchanges hairstyle tips with Andy Murray.
The men discuss curls, smoothness and "being worth it" over Twitter.
The sport is no stranger to funky hairstyles over the years.

The model generated summary:

(CNN) What does 17-time Grand Slam winner Roger Federer have to talk about with World No. 1? I saw you checking your hair out during the photo shoot yesterday #silksmooth, before posting an image of himself from 2008 with a curly mane, adding "let me know if you ever want to go down this route." Earlier this month, Federer suffered some embarrassment of his own when a small boy played a point against the 33-year-old -- outmaneuvering the Swiss legend with a perfect lob at an exhibition match in New York.

Screenshot 5:

The Reference summary:












Amran Abdundi wins the campaigning category at 2015 Index Freedom of Expression awards.
Honored for her work helping victims of rape and conflict along Kenya-Somalia border.

The model generated summary:










The ongoing threat from the Islamist militant group Al-Shabaab means that remote villages and citizens suffer indiscriminate attacks. We go there and we advise them to go to school, advise their parents against forced marriage, tell them to take the children to school, stop circumcision. AA: Yes, I would hear there are some gunmen or they want to do something around the villages, or maybe they want to attack the villages and destroy it or take livestock.

Rouge Score for Text-Rank Model

Rouge-1:

Python Console	Python Console(2)	Python Console(3)	Python Console(4)	Python Console(5)
 2  3  4  5  6  7  8  9  corpus length :800  Rouge-1 score for TextRank summarization is:0.9508440757853693  In[3]:				

Rouge-2:

Python Console	Python Console(2)	Python Console(3)	Python Console(4)	Python Console(5)
 4  5  6  7  8  9  corpus length :800  Rouge-2 score for TextRank summarization is:0.21576348780188423  In[3]:				

Based on the results above, it is clear the rouge scores are not very different between the models. This is surprising because the first model used a simple TF-IDF vectorizer to compute TF-IDF scores in the training corpus to generate summaries for new news articles while the second model was a lot more complex with complex math to weight the edges in the graph and compute the final vertex scores and sentences for the predicted summary. We rationalize this to be due to the same TF-IDF vectors being used by both models to compute the summaries.

Nevertheless, it is important to note the significant drop in rouge score going from unigram to bi-grams across the models. Using unigrams models yielded a rougescore of about 0.95 but bigrams yielded a score of around 0.21. This may be attributed to the fact that the reference summaries we had from the CNN dataset were abstractive summaries, where human readers composed a short summary of the article in their own words. In the process, they seemed to incorporate a lot more new words and phrases in their summaries in comparison to the actual article. For instance, human writers had phrases like “The 15 new cardinals will be installed...” and “No Americans made the list this time or the previous time in Francis’ papacy”, when the original article never had phrases like “will be installed” or “this time or previous time in Francis’ papacy”. Our summarizers worked only with the original articles and hence were unable to produce summaries that match with the references.

Summary

Text summarization is a very popular and intensively researched area of natural language processing especially since the early 90’s. Given the limitation of machines unable to intuitively form a semantic representation of natural language unlike human counterparts, there is a strong need for effective summarizers that capture all the important and only the important aspects of input data. Numerous researchers have proposed even greater number of techniques to improve several state-of-the-art systems such as simple feature-based models passed into naive bayes classifiers and graph-based techniques like TextRank to complex models attempting to form semantic representation of ideas and concepts via centroid-based extraction, machine learning, and even deep learning.

In our project, we got the opportunity to examine a naive statistical model using TF-IDF and a complex model namely TextRank. Although the models are very different in their approach and complexity, it is very surprising to find them both perform very similar to each other in terms of accuracy and rouge score. This tells us that a model need not really be complex to perform better than its simpler counterpart. One interesting finding however is our models seemed to perform equally well on the small dataset of five articles as with a much larger dataset of 3,200 articles. This is a good indication that our models are not overfitted at least.

Conclusions

As we were all new to the field of natural language processing, this project gave us a great opportunity to learn and work with basic NLP concepts like data preprocessing, unigram/bigram approaches, and model fitting. Even if our models turned out to be effective for unigrams, it is not up to the mark compared to some of the state-of-the-art systems. This can be attributed to our newness to the field and to the python programming language. Looking back, one thing that can be improved upon in our TF-IDF model is using POS tagging and weighing more features like proper nouns, sentence length, and sentence position in the original article. Also, we could implement a hybrid model combining both

TextRank and feature-based approach by computing the edge weights to account for additional features as well. After reading over 20 research papers in this field as a team, we feel this would be a very unique idea worth trying.

Although we faced a lot of challenges in understanding the math behind some of the algorithms and with python syntax, overall this project was very enlightening. Not only we had gained lots of hands-on experience researching and reading various kinds of nlp papers, but we also feel a lot more confident working with various machine learning libraries in python like scikit learn and nltk, and interpreting results and performing analyses on our findings.

On the broader picture however, this project showed us how complex the whole field of NLP is and how it is still in its infancy in many ways. Even with all the research and numerous findings published each year, it is clear natural language is not as straightforward to computers as it is to humans. It is very difficult to “teach” computers to form a semantic representation of even simple objects like cars and toys, let alone complex ideas and relationships. In our case, it was challenging enough to compare and quantify how our summary matched with the provided abstractive summaries. It would be even more challenging to “teach” our summarizer to produce similar abstractive summaries.

References

1. NLP Progress website
<https://nlpprogress.com/summarization.html>
2. A survey of text summarization techniques
https://link.springer.com/chapter/10.1007/978-1-4614-3223-4_3
3. COMPENDIUM: A text summarization system for generating abstracts of research papers
<https://www.sciencedirect.com/science/article/pii/S0169023X13000815>
4. Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)
<http://journal.binus.ac.id/index.php/comtech/article/view/3746>
5. Top 4 online text summarizing tools
<https://www.maketecheasier.com/5-useful-tools-to-summarize-articles-online/>
6. TextRank: Bringing Order into Texts (Rada Mihalcea and Paul Tarau)
<https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf>
7. Google's PageRank algorithm with the damping factor
<https://web.stanford.edu/class/cs54n/handouts/24-GooglePageRankAlgorithm.pdf>
8. PageRank as a Function
<http://vigna.di.unimi.it/ftp/papers/PageRankAsFunction.pdf>