Creating a Document Summarizer for Novices

By

Rex Rubin

* * * * * * * * *

Submitted in partial fulfillment
of the requirements for
Honors in the Department of Computer Science

UNION COLLEGE
March, 2018

Abstract

REX RUBIN Creating a Document Summarizer for Novices. Department of Computer Science, March, 2018.

ADVISOR: Aaron Cass

I am looking to see if adding a glossary to a summarized document will extract more coherent sentences to the final summary. Getting into a field of research is daunting with research papers giving a lot of difficult information, I am looking to extract the easiest to read sentences for those new to the field. To do this, I will be editing an already existing Python program to include a glossary of words related to the original document. I currently have a working version of the summarizer, and the documents that will be run through it, but testing the effectiveness of the augmentation has not yet begun.

Contents

List of Figures

List of Tables

1 Introduction

My original question was how could I improve a text summarizer to make it easier to read for all students. It was changed to ”Would adding a glossary to the original document bring out more coherent sentences for beginners in a field.”, earlier in the term. I believed the best way to venture into a new field would be to read several papers specifically on points in the field. I pursued this problem, as everyone has been at the point of starting a new field of study, and reading endless research papers is very difficult as most of the information can seem disheartening. The goal of my summarizer is to find the most important parts of the paper to present to the user, as they should give the best description of the paper overall. This is not to say that the rest of the paper is unimportant, but most of the information will not make any sense to the uninitiated mind of a beginner.

This is important because everyday there are people being introduced to new fields, many of which take a glance and determine that it is not for them based on reading papers alone. I want to make the process of joining a field easier so that those who would turn away are given a better introduction before making their final choice. A text summarizer makes sense in extracting the important information in a text, but what about when the information given does not make any sense to the reader? Simply giving what the summarizer thinks to be the most important sentences does not help the overall problem, as said sentences could easily be just as confusing as the rest of the document.

There are already dozens of different text summarizers though, so what about mine would be different? I will be adding a glossary to the original text as a way to bring out more insightful sentences in the final product. The way most text summarizers work is by determining the most important sentences by comparing them to the other sentences. If there were different sentences in the text, the comparisons would also be different. This should cause a different set of sentences to be chosen for the final product, hopefully with more coherent information to the subject. A normal summarizer may extract what it believes to be the most information filled sentences to present, but these sentences may be very confusing to someone with no previous knowledge. The glossary must also be given by the user, which should also mean that they are familiar with the contents of it. This is where the specific modification I have made gets more specific. A research paper is not normally meant to talk about the entire research field, but certain points in it. My summarizer will take a glossary of terms related to the research paper given in order to extract the most important sentences.

Why specific points of study? In order to become more aware of a broad field, smaller portions of it must be inspected first, you cannot complete a puzzle without analyzing each smaller piece. Along with this, it allows for the glossary to try and bring out specific sub-topics in the sub-topic. For example, if my summarizer were to be used on a research paper about document summarizers, if the glossary was specifically filled with terms about extraction, the final abstract would have more sentences about extraction, narrowing the already narrowed research paper. Although its purpose is to see if adding a glossary extracts more important sentences for a novice, it can fulfill other side purposes. My question now is ”How will adding a glossary to the original document affect the comprehension of the material by novices to a field.”

2 Background and Related Work

To start making a document summarizer, it was very important to first research the field a good amount. Doing this taught me about Extraction-Based and Abstraction-Based summarizers.

Extraction: Extraction based summarizers work by comparing sentences and seeing the similarity between each of them [3]. It tries to find a relation between the words as a subset and evaluates them. Once all of the sentences are evaluated, a score will be given to each based on the information relevancy. The top scoring sentences will be chosen for the final abstract made. The drawback from using this method is usually the coherency of the abstract [3], as the sentences used for it are the exact same as they were in the paper with no modifications made to it. What this can cause is some information to be presented poorly, with a bad transition in between sentences, being able to shift topics rapidly. However, this method is better for summarizing single documents, as in single documents there is usually not enough information to form new computer generated sentences, which is how Abstraction based summarizers work.

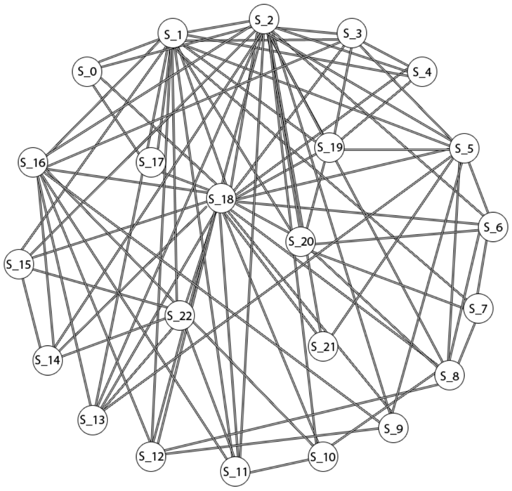

Figure 1: A sample TextRank graph

Abstraction: Abstraction based summarizers work by comparing sentences throughout the document or documents, and will try to condense them into sentences that contain more information [1]. It will still score sentences and such, but it will reduce the overall amount of sentences total causing there to be less to choose from in the final summary. This method also works much better for multiple documents around the same issue, as it can condense information through the documents. It is also available for single documents, however it works much better for multiple documents, as the more information it has to go off of, the better the final abstract should be [2]. Compared to Extraction, this will produce more coherent abstracts as the sentences are constructed to have a certain flow to them, as they are not just assorted sentences from the original document, but sentences meant to transition into each other.

I chose to modify an already existing document summarizer, as there is no need to make what already exists. My base was a Python program that ran a well known extraction based summarizer, TextRank [4]. TextRank works by forming a graph out of the document (refer to figure 1), having all the sentences be nodes that would connect to each other. They are connected by vertices which determine their similarity in words. The vertices are weighted, which is how the sentences are scored. Based on the weight of each vertex, a score is given to a sentences. The weight is the similarities between the sentences, determined by the author [6]. In this case, the weights are decided by the amount of words shared in a sentence. Once all of the vertices are weighted, the graph will be inserted into PageRank.

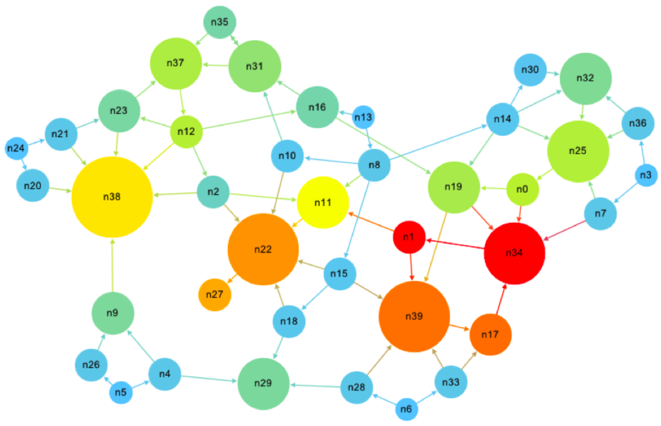

Figure 2: A sample PageRank graph

PageRank is the system Google uses to determine the sites that are relevant during any given search. How PageRank works is it will take the words in the sentences, and determine their relevancy to each other, based on several factors such as placement in a sentence, length of a sentence, words shared by other sentences and so on [5]. This will then help determine another graph similar to TextRank, except using words over sentences as nodes (refer to figure 2). The whole purpose is to determine the importance of each word, and find the main subject of the paper through it. PageRank will then use the words to score the sentences sent in through TextRank, and then rank the best scoring sentences [5]. The best scoring sentences will then be used for the final summary, appearing in the order they were originally presented on the original document to preserve coherency [4].

3 Methods and Design

I chose to use and extraction based summarizer for various reasons. Abstraction is considerably more difficult to work with as it requires much more knowledge in natural language processing. I originally thought that using data mining techniques would work better for creating a summarizer, but this created many complications. Changing around the scoring algorithm with adding different sub-algorithms does not help the final result, but can skew the results [2].

Second, the amount of text I plan for the summarizer to take is significantly less than what is desired by abstraction based summarizers. Abstraction is really meant for many large bodies [1] of text which is the opposite of what I am trying to use my summarizer for. I want the novice mind to journey through many different fields to gain a broad knowledge through specific points. Incredibly large bodies of text is the last thing that the demographic this summarizer is aimed for should be reading. My summarizer does fail in the sense that it will normally give a worse summary the larger the research paper is, but the point is for beginners just getting started who will most likely not read a very large research paper for a while.

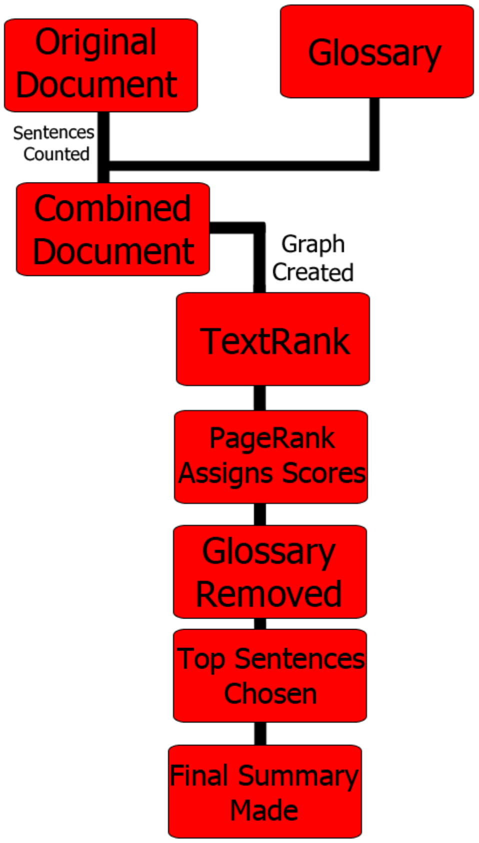

Figure 3: Planned modification for TextRank

The last point is the most important, being that an abstract summarizer cannot use the glossary as well. The way an abstract summarizer works is by taking all the sentences and condensing them, essentially making the glossary unable to be removed at the end. It could make the sentences bring out more information, but there is also a chance that it would return mostly a differently phrased glossary, which is not what my plan is. The glossary should be affecting the sentences chosen, but it should not be a part of the sentences, because at some point they may just be reading the glossary again. The glossary should normally have a similar size compared to the original document which would make the ratio of glossary sentences to non-glossary sentences too high.

My plan going into modifying TextRank was to take the original document and the glossary and put them both as inputs into TextRank (refer to figure 3.). From there I wanted TextRank and PageRank to score the sentences and prepare them for the final abstract. The glossary sentences would then be removed at this point as to make sure they were not chosen as any of the top sentences. The top sentences would then be taken and put on the final abstract. In this process, the summarizer will already be placing the sentences in the order they were in on the original document as to avoid any confusion with scattered data.

In doing so, there was a function that would index the functions by the order they were presented. Each sentences obtained an ID in the form of an index number, and the summarizer would take the sentence from the index to place it in the final abstract. This way of IDing sentences was also how the glossary is removed at the end. Because the glossary starts at a certain index number, before the final sentences are chosen, I modified the code to remove the parts of the index after the original document. This allows it so no glossary sentences will be chosen, as they will all be removed before the process of choosing sentences. It is important that the glossary is not to be removed before the sentences are given scores, as this is the part where the glossary will improve the scores of sentences better suited to novices.

There is already an input in the summarize function that determines how much of the original document will be returned as the final abstract [4]. I chose not to put in a word limit, as that may hinder the knowledge given back by the final abstract. The purpose is not to only present the most important sentences, but it is mostly to trim the excess sentences from the original document. It is not uncommon for things to be repeated throughout the whole paper, so I would like to remove any extraneous sentences, not just shorten the paper. I have it currently set to return 20% of the original document, as I believe this provides a good amount of information in it without making the abstract too long.

When deciding on a test document to use, there are several factors to account for when running the program. The document must be specified on a certain subject, as the glossary must be changed to match the subject. If it is a very broad document on the entirety of Computer Science, the glossary will be much less effective. The test summary that I created focuses on Cybersecurity, as it is a specific enough point to focus on. The glossary for the Cybersecurity document was heavily filled with terms exclusively about Cybersecurity, as to try and emphasize the changes made. The glossary of Cybersecurity terms totals up to 302 terms and definitions, in hopes of having them have matching words with the original document.

4 Experiment

I had 18 total participants in my experiment, all of which were Union College students. They were randomly separated into 2 groups, being the Control Group, and the Test Group. I made the group selection randomized by having the first participant be part of the Test Group, and the next participant be part of the Control Group. The differences in the Control Group and the Test Group were:

Control Group: The Control Group read through the summary created by the original unmodified version TextRank.

Test Group: The Test Group read through the summary created by the modified version TextRank that included the glossary as an input.

4.1 Designing the Test

I used an article focusing on Cybersecurity and the usage of the SCADA system, and how it could be tampered with. SCADA stands for Supervisory Control and Data Acquisition, a system that is used to monitor other systems in a network. This article focused largely on the flaws of the system and how one could exploit them. To judge the efficiency of the summarizers, I created a test filled with questions on what I believed to be the most important points of this article. I went over several ideas of experiments that could accurately judge the levels of comprehension in the participants, and concluded that a test on the article would work the best. A test challenges the mind of the reader, needing to answer questions on the subjects, whereas other forms of judging their assessment fall short. I had considered giving the participants a survey about how well it conveyed the points of the article, but the results of this would be largely subjective, as well as the purpose of the summarizer is not to make it easier to read, but easier to find key information. Both of the groups received the same test, as both of the summarizers should return the main points of the article. Neither of the summaries had been created at the I created the test to remove any potential bias.

There were 6 questions, 3 multiple choice, and 3 open answer. The first question was a simple open answer question asking what the main topic of the article was, the answer I was looking for specifically was Cybersecurity. The second question was an open answer question, but the answer needed to be inferred from the information given in the summary, as it was asking what SCADA meant. The first multiple choice question was the third, asking which product had microprocessors built into them, as this was discussed in some length in the article. There were 4 possible choices, the 3 other options were devices I made up scrambling words used for other devices that were mentioned briefly in the article. Following this was another multiple choice question, this one with only 2 options. I asked whether the article was highlighting problems caused by transmissions through wi-fi, or Ethernet, the answer being Ethernet, as the article discusses how the system is connected through Ethernet. The last open answer question, question 5 asked what category of attacks were more common before 2001, and then after. The article discusses how before 2001, most security breaches were physical break ins and tampering with equipment, while after 2001 they were much more Internet based attacks, coming from people who never left their homes to tamper with the system. The last question was a multiple choice question, asking which of the following networking devices was not investigated as a precaution. For this I gave 3 of the things it mentioned were checked, and used the same method to create a fake device as I did for question 3.

Using a test instead of another form of assessing the participant did create several implications, the first of which is that previous knowledge could skew the results. I chose to use both Computer Science students and non-Computer Science students for this, despite it being a Computer Science article. To compensate for this, I used a topic that was for the most part new to all of the participants. There had yet to be a Cybersecurity course at Union, until the Winter of 2018. CSC 483 was a course focusing on the applications of Cybersecurity, however it was an upper level course, so not as many students were taking, making the chances for a student with previous knowledge on the subject to be low. The whole purpose of this was to try and make each participant start with a minimal amount of knowledge, so they could be considered beginners to the subject.

4.2 Administering the Test

I had the participants read and sign an Informed Consent Form, as required by the Union College Human Subjects Review Committee, informing them of any foreseeable risks and that they were able to opt out of the experiment at anytime, and that their results would be kept anonymous. Once they had signed, the participants that were chosen for the Test Group were given the summary created using my modified version of TextRank, while the Control Group was given the summary created by the original version of TextRank. The summary of the original article was shortened from 10 pages to half a page. I then gave the participants the test, and asked them to answer the questions to the best of there ability. The participants were not told of there being two separate groups, or what the purpose of this test was until they had finished. The participants were given as much time as they needed to finish the test, as not to rush them and answer differently based on the time constraints. When they finished, they were told the purpose, why they took the test, and to not tell anyone else about the contents of the experiment.

5 Results

Of the 18 participants, there was a mix of Computer Science students, and students uninvolved in Computer Science, however I do not believe this caused a huge gap in the data. The average scores of the data were very close for both groups, other than questions 4 and 6. Of the results, I was very surprised with question 2, as I believed this question would be very easy, as question 2 is asking for the specific main point of the article. To get a better look at the averages though, I needed to remove and potential outliers. I found 3 total outliers, which I had identified by checking if the scores fell outside p-values range. These 3 outliers fell far outside the norm for several questions, which did skew the data.

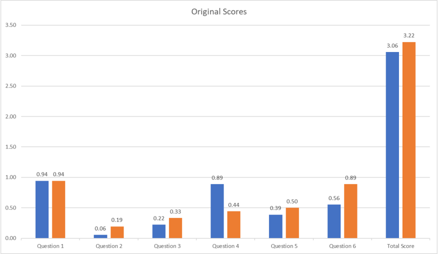

Figure 4: The averages of both groups test scores. Blue: Control Group, Orange: Test Group

With the outliers removed, there was a larger gap in the data for the Control Group compared to the Test Group as can be seen in figure 5. The average of the total score in the Control Group lowered to 3.00, and the Test Group raised to 3.63. To then check if my hypothesis was correct, I needed to check if the difference was large enough to be considered significant. The difference of the total scores was around 1/12th of the test's total points, and something that I figured would be considered significant. I checked if the difference was significant by checking whether the difference was outside the p-value range, and if it was outside 2 standard deviations of the average. My hypothesis was incorrect however, as the difference was inside both of these values.

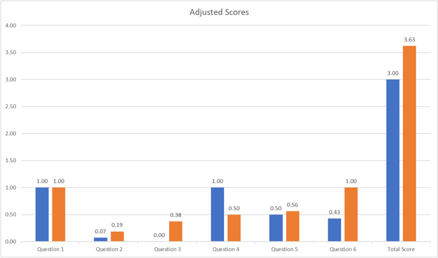

Figure 5: The averages of both groups test scores with outliers removed. Blue: Control Group, Orange: Test Group

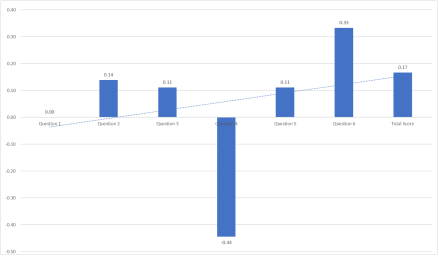

Figure 6: The differences in average scores. Calculated as Test Group - Control Group

To understand the reason why this was not significant enough, I had to more closely analyze the data. Something that I noticed was that the standard deviation for both groups was too large, as the data did have a large range. Many scores were either very high, or very low, giving a larger range for the standard deviation. I believe this is caused by having a mix of Computer Science students and non-Computer Science students. There were several answers in the tests that would leave the impression that the participant had no idea the article was even talking about Cybersecurity, as this could be seen by the answers to questions 1 and 5. There were very few answers to question 1 that could be considered wrong, but of those that were, the answers showed that the participant did know it was about security, but not necessarily computer security. For question 5, it asked as to how attacks to the system were occurring, and a couple of answers focused on only physical attacks, unlike the digital ones described in the article. The differences in the average scores between the two groups was not significant enough to justify one of the summaries conveyed this information better than the other however.

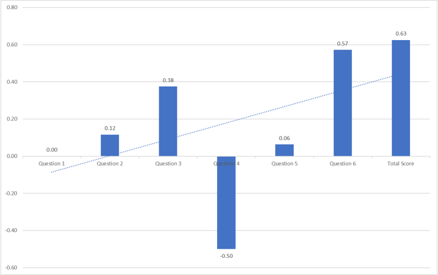

Figure 7: The differences in average scores with outliers removed. Calculated as Test Group - Control Group

Although most of the questions had relatively the same score, 2 questions had very different average scores. Questions 4 and 6 had differences significant enough to justify further examination, with the difference in question 4 being 0.50, and 0.57 for question 6, with the outliers removed. I had figured that the multiple choice questions would be relatively similar between both groups, but I was sorely mistaken. Only one participant answered question 4 wrong in the Control Group, while less than half of the Test Group's participants answered it correctly. To make sense of this, I read through both summaries while specifically looking for this topic, and I discovered that it was communicated much better in the Control Group summary, while the Test Group summary had barely anything on the matter. Since it was multiple choice and had only 2 choices, it makes sense that the average for the Test Group was averaged to 0.50 with the outliers removed, as this is similar to flipping a coin in their case. For the Control Group, since they had much more information on the question, it makes sense that they would have a much higher average score.

On the other side, the Test Group's summary had much more information on question 6, which focused on the devices that were monitored as a precaution, asking which of the listed devices was not monitored. I believed that with a little background in this topic, the participant would be able to correctly answer this question, as I began thinking that the question might have been too easy. The differences in the two groups clearly show that this problem was not too easy, as the majority of the Control Group answered this question incorrectly. Checking the Control Group's summary for this information, I found little to nothing

Term	Course	Aims/Outcomes
Junior Spring	CSC497	Find Advisor, develop ideas **A research question and fully formed proposal**
Senior Fall	CSC498	develop several algorithms designed for research papers create a Python application using my algorithms **Have a fully working Java application of my summarizer**
Senior Winter	CSC499	test my summarizer on students determine if it works or not **A Completed Capstone Design Project**

Table 1: A time line for a CS Thesis

on the subject, which was surprising that the average score was around 0.50. Checking the summaries to see the differences in the scores did lead me to find something else about the summarizers.

Extraction based summarizers are far from complete, as both of the summaries were difficult to follow along with, and presented the information in a somewhat confusing manner. The purpose of the summaries are to present the most important information in the original article, but the two displayed different information, this is not to say that the information they displayed was not important however. I believe that the size restraint of the summaries is what limited the information that was shown to both groups. Abstraction based summarizers are successful at making summaries, but only with large sets of data to work with. As Extraction based summarizers are meant to work with smaller documents, they cannot form new sentences, thus they need to determine what is more important. It is clear by both summaries that they believed both points were important, but there was not enough room to express this . The next thing that is important about this, is how each summary had different main points. I do not believe that my modification to TextRank was inferior to the original, but they both had different ideas of what was important.

6 Future Work

With more time, I believe that my hypothesis could still be possible, but in this set of data, it is not. With a larger sample of participants, I would be able to obtain more definite results. Along with this, more articles need to be tested with the modification, as this would also give a larger range of data to analyze. Along with this, it would be nice to try the experiment on specific demographics, as in only Computer Science students, and only non-Computer Science students, as to see if having a mixed group caused significant standard deviation. I also believe it to be worth pursuing more in depth as to why the modified summary presented different information, and how much that does change when trying a different glossary of the same subject.

References

[1] Kathleen McKeown Dragomir R. Radev, Eduard Hovy. Introduction to the special issue on summarization. *Association for Computational Linguistics*, 2002.

[2] Magorzata Stys Daniel Tam Dragomir R. Radev, Hongyan Jing. Centroid-based summarization of multiple documents. *Association for Computational Linguistics*, 2003.

[3] Jan Pedersen Kupiec, Julian and Francine Chen. A trainable document summarizer. *ACM SIGIR conference on Research and development in information retrieval*, (15):68–73, 1995.

[4] Frederico Lopez. Textrank implementation in python. 2017 (last update).

[5] Herwig Unger Mario Kubek. Topic detection based on the pagerank's clustering property. 2011.

[6] Paul Tarau Rada Mihalcea. Textrank: Bringing order into texts. 2011.