Automatic Document Summarization

Nitin Pathania

December 3, 2020

Abstract

Identifying and summarizing relevant information in a given document is a task that can help reduce the time required for acquiring and acting on new information. Accurate summaries can reduce reading time and personalized summaries could help a reader better understand material they care about most. The work described in this project identifies relevant recent advancements in the field of **Document Summarization**.

1 Project Overview

Automatic Document Summarization is the task of rewriting a document into its shorter form while still retaining its important content. Every day we are bombarded with more information than is possible to process. Identification of the most relevant information in a given document is a task that can help reduce the time required for acquiring and acting on new information. Accurate summaries can reduce reading time and personalized summaries could help a reader better understand material they care about most.The most popular two paradigms are extractive approaches and abstractive approaches. Extractive approaches generate summaries by extracting parts of the original document (usually sentences), while abstractive methods may generate new words or phrases which are not in the original document.

There has been significant work in the field of Document Summarization over the past two decades. In the early 2000s, a set of conferences named DUC (Document Understanding Conference) focused on text understanding and summarization and provided many of the standard data sets still used today. Researchers such as Daniel Marcu and Inderjeet Mani have written several books and academic papers on the subject detailing machine learning approaches. Recently, the Google Brain team have seen some improvements with their Transformer and XLNet based systems[1] and additionally Microsoft Word at one supported basic automatic text summarization but has since removed this feature.

My personal motivation for studying this topic is its relevance to Natural Language Understanding. Identifying the most salient points of an article or

[1] https://github.com/zihangdai/xlnet

chunk of information is one step of the way toward automatically *understanding* the information an author is attempting to convey. Document Summarization also involves many areas of NLP and requires solving several subproblems (e.g. parsing, entailment, entity detection, etc) in order to achieve good results which also interests me.

1.1 Problem Statement

This goal of the project is to write a program that will take a piece of text, which could be an article or a review, and generate a summary automatically. The program will process the document with a trained model and return a short summary of the most important topics from the document. The summary could be a sentence from the document itself or potentially a set of phrases from the document that best encapsulate the central theme of the document.

1.2 Metrics

The standard metric used for measuring model performance in document summarization is the ROUGE score[2]. This is what I have used for quantifying the performance of the deployed models. The ROUGE score is a measure of the n-gram overlap between the model generated summary and a gold standard summary. There are different variants of the ROUGE score, for example, ROUGE-2 is a measure of the bigram overlap between the machine and human generated summaries. ROUGE-S is a measure of the skip-bigram overlap, where a skip-bigram is any two words that appear in their sentence order from the original text.

Alternative, or additional metrics could include measuring the noun phrase chunk overlap between the machine and human summaries (really this is a variant of ROUGE) or the Levenshtein distance[3] between the machine and human summaries.

2 Analysis

2.1 Data Exploration

2.1.1 Datasets

The ideal input for this project are news articles. No particular domain of news articles is specified for this endpoint though I expect there to be differences in performance that can be attributed to differences between article domains (e.g. sports vs tech).

For training the model, there several available datasets. The DUC 2003 and DUC 2004 sets are the standard sets that many research uses as a baseline. The

[2]https://en.wikipedia.org/wiki/ROUGE_(metric)
[3]https://en.wikipedia.org/wiki/Levenshtein_distance

DUC was created specifically for the conferences and research into question answering and document summarization specifically. The 2003 and 2004 datasets are the ones most often used in summarization work, with the 2003 used for training and the 2004 set used for testing. The DUC dataset is general news stories which is a good fit for the problem.

For additional training on news article domains there is the CNN/DailyMail data set that is freely available[4] as well as the Cornell Newsroom dataset[5]. These would be good additional training data or testing materials.

2.1.2 Benchmarks

There are several models described in [1] which could be used a reference for the performance of the model I have built for this project. The *FreqSum* model is simple to implement (simply being a measure of the approximate importance of words in the input based on their frequency.)

3 Methodology

The initial steps of this project was to acquire and analyze the existing data. For the DUC data there were permission and usage forms to fill out and send to NIST. For the other data mentioned above, it was the need to be downloaded and placed in a location that will make exploration of the data possible with Python. I have already acquired most of this data and got DUC data on NIST after submissions for not sharing data anywhere else for the DUC datasets.

The type of analysis that the data needs will likely be tokenization and basic NLP manipulation such as named entity, parsing and part of speech tagging. There will also need to be unigram, bigram and potential larger n-gram counts to collect as well as term frequency and inverse document frequency values. Stop word removal will also be needed. This can all be collected and implemented outside of the main model development though it's likely code from these steps would also be leveraged in the main model project. Likely the Python Natural Language Toolkit will be used to take advantage of the tools within in it that do much of this work and manipulation already.

A format for summaries needs to be selected that will allow for easily scoring with the ROUGE metric and output by the systems to be built. Potentials are raw plain text or JSON.

A script to score machine generated summaries via the ROUGE metric needs to be implemented. This will be done in Python 3 and included in the project. There are implmentations of the ROUGE metric available on researcher's pages and Github accounts that are available for free use.

The *FreqSum* system needs to be implemented so that is may be used as a baseline system. This will be done in Python 3 and included in the project.

[4] https://cs.nyu.edu/ kcho/DMQA/

[5] https://summari.es

The *RegSum*[2] system needs to be implemented in an initial state for quick local development and iteration. *RegSum* utilizes weights from 3 unsupervised approaches to summarization as well as features drawn from standard Natural Language Processing data manipulation approaches such as Named-Entity Recognition and Part of Speech tagging.

RegSum incorporates features derived from Log-likelihood Ratio tests developed in[4].

Once the baseline model and the RegSum+ (RegSum + any changes and additional features I implement during the data analysis) are implemented the RegSum+ model will need to tuned. Many variations of the feature set and hyperparameters will need to be tried in order to improve the results of the model.

3.1 Preprocessing

There is substantial amounts of preprocessing required for document summarization. This project relies on multiple other NLP techniques such as tokenization, Named Entity Recognition and various forms of word similarity.

3.2 Results

In Table 1, the results for the baseline *FreqSum* system are provided on the training set of DUC2003 documents. Since *FreqSum* is unsupervised we are able to present scores for the delegated training set as well as the testing set of DUC2004. Those results are found in Table 2.

Metric	Recall	Precision	F-Measure
ROUGE-1	0.2570	0.2311	0.2419
ROUGE-2	0.0475	0.0452	0.0512
ROUGE-L	0.2223	0.2009	0.2072

Table 1: *FreqSum* results of DUC2003

Metric	Recall	Precision	F-Measure
ROUGE-1	0.2669	0.2495	0.2569
ROUGE-2	0.0666	0.05888	0.0622
ROUGE-L	0.2327	0.2177	0.2219

Table 2: *FreqSum* results of DUC2004

The first implementation of the *RegSum* system only contained the *topK* feature set. It performed noticeablely worse than the *FreqSum* system as documented in Table 3.

Metric	Recall	Precision	F-Measure
ROUGE-1	0.1980	0.1833	0.1897
ROUGE-2	0.0262	0.0254	0.0256
ROUGE-L	0.1768	0.1634	0.1678

Table 3: *RegSum* topK results for DUC2004

4 Conclusion

We are able to create base models for freqSum and RegSum models.

4.1 Future Work

In the future stage we can conduct these experiments utilizing the AWS or any other cloud. Deploying the model to AWS and SageMaker can happen during this time as well. There will be a need to create a model available in AWS as well as create an endpoint that can be accessed outside of AWS. A simple webpage that can call the endpoint and supply a document to it and receive the summary should also be created in future scopes.

References

[1] Kai Hong, John M. Conroy, Benoit Favre, Alex Kulesza, Hui Lin and Ani Nenkova, *A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization*, Proceedings of LREC, 2014.

[2] Kai Hong and Ani Nenkova, *Improving the Estimation of Word Importance for News Multi-Document Summarization*, Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014.

[3] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown, *A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization*, Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006.

[4] Chin-Yew Lin and Eduard Hovy, *The automated acquisition of topic signatures for text summarization*, Proceedings of the 18th conference on Computational linguistics, 2000