Text Summarization Tool Evaluation: A Study On Automatic Summarizing News Articles

Michael Foley, Andrei Gerashchenko, Rachael Tovar, Zhou Tong
Department of Computer Science
Wheaton College
Norton, MA
{foley_michael, gerashchenko_andrei, tong_tony}@wheatoncollege.edu

1. INTRODUCTION

In the past decade reliance on data-driven decision-making has grown significantly and in turn, so has the need for reliable and efficient data-driven intelligence collection. The challenge of combing through massive amounts of data and producing meaningful insights has arisen. It is becoming increasingly evident that manually processing, digesting, and summarizing large amount of documents within a reasonable amount of time is either cost-prohibitive or unpractical. Therefore, we resort to automatic text summarization to help combat this challenge. Text summarization has been a developing field for almost 40 years and various extractive and abstractive tools have been developed. This paper aims to conduct a feasibility study of incorporating eight popular text summarization tools to generate text summaries in a single framework. A preliminary performance evaluation is carried out based on a collection of 60 news articles.

2. BACKGROUND

Text summarization is the creation of a shortened version of a text that outlines the input text by a computer program. The output of this process is referred as a summary. Knowing how to quickly extract intelligence from a large document is helpful in both academic and business sectors. News editors sum up news or stories for a good headline to attract attention. Lawyers summarize documents for court hearings and legal proceedings to support their cases. Business marketers analyze product reviews to extract general sentiments to make better marketing decisions. In other words, the need to read, process, and analyze large quantities of documents is ubiquitous. However, summarizing documents manually is an arduous process. In this fast-paced society where intelligence is expected to be readily available, it is extremely challenging to produce quality summaries on demand. Therefore, in this study, we survey various third-party text summarization software and present our preliminary evaluation results on the efficiency and accuracy using a collection of online news articles.

There are two different types of text summarization approaches, abstractive and extractive. Although both approaches exploit the use of natural language processing and statistical methods to generating summaries, abstractive approaches are usually much more computationally intensive and can produce summaries that are more condensed than the extractive approaches.

Extractive models extract sentences that are deemed important by the algorithms from the input text directly and output them as a summary. Extractive methods usually use a ranking algorithm to determine the significance of each sentence and rank them from the most to the least important. Since no additional words are added, it is common to see that an output summary is produced by reorganizing important keywords that focus on a topic and removing transition phrases, unnecessary clauses, and excessive examples from the source [9]. Therefore, for extractive tools, the quality of the summary depends on whether it can find the most important sentence that carries the most information. The main limitation of this approach is that the output summary may not be conclusive if such sentence doesn't exist in the original text.

In contrast, an abstractive model needs to understand the topic first and then summarize the input text to create a shortened text with its own words or paraphrasing sections of the source document. It is considered more challenging to develop than the extractive models as it is closer to how humans read, process, and summarize a text document. To create a more cohesive summary, it relies on advanced natural language processing tools and machine learning models. To achieve the best result, it usually requires a model training process that builds the model based on a large pool of domain knowledge. Thus, it generally requires larger computing power and a longer processing time. Abstractive approach is more favorable when the input text is complicated in nature. One of the main challenges for the abstractive method is to produce a comprehensible summary with desirable grammar that human readers can not distinguish from a user-generated summary.

The rest of the paper is structured as follows. Section 3 discusses the related works. Section 4 introduces our framework that can summarize news articles by multiple text summarization tools. Section 5 presents our preliminary results on the accuracy of select text summarization tools with 60 articles. Lastly, Section 6 concludes the paper with future direction of research.

3. RELATED WORK

Text summarization or automatic summarization has been around for many decades. A large number open-source ranking-based and machine learning based text summarization tools have been published and evaluated by various evaluation metrics in the past. One of the most popular scoring metric ROUGE is developed by Lin et al. in [4]. Gambhir et al. [2] reviewed recently published text summarization tools using ROUGE-1 and ROUGE-2 evaluation methods and found that the top methods were Progressive and Ranking-based MMR techniques. Moratanch et al. [7] surveyed extractive text summarization techniques and used both human and ROUGE evaluation metrics. Allahyari et al. [1] analyzed various single and multi-document text summarization approaches including topic representation, frequency-driven, graph based, and machine learning approaches and their shortcomings. They found that utilizing topic representation is both effective and accurate for multi-document summarization, but when ranking the importance of a sentence, it tends to favor longer sentences. Sanhet et al. [8] proposed a general purpose light-weigth model DistilBERT that is 60% faster and 40% lighter than BERT but retaining 97% of the language understanding capabilities. Liu et al. [5] developed ReBERTa which is improved version of BERT as they found that BERT was under-trained. With more training, ReBERTa achieved better performance on GLUE, RACE and SQuAD NLP benchmarks. Lewisal et al. [3] developed BART a de-noising auto-encoder for pre-training sequence-to-sequence model. The performance of BART is similar to RoBerta in discriminative tasks and multiple other state-of-the-art text generation tasks. Miller et al. [6] described that the extractive BERT model have similar weakness to other extractive tool such as difficulty handing conversational words and context words. Zhang et al. [10] proposed PEGASUS a pre-training transformer-based encoder-decoder models. PEGASUS achieves great performance on multiple data sets based on human evaluation.

4. THE FRAMEWORK

We developed a framework that can summarize news articles with multiple text summarization tools in a single run. In this section, we introduce the summarization tools used in this study, then we discuss the evaluation metrics. Lastly, we use a sample news article to demonstrate how our framework can generate, compare, and evaluate various machine-generated and human-generated summaries in a web application.

4.1 Text Summarization Tools

We have managed to configure and execute eight well-known text summarizing tools that are publicly available online. Table 1 lists all eight tools with their summarization type, developer, description, and training status. SMMRY, Sumy, Textteaser, and Bert are based on extractive models. The other four tools are abstractive and mainly use deep learning methods to generate summaries. We decide to use the pre-trained models for this study.

4.2 Evaluation Metrics

In this study, we use the Recall-Oriented Understudy for Gisting Evaluation for unigrams (ROUGE-1) as it is the most widely-used evaluation metric that automatically evaluates the accuracy and similarity between the machine-generated text summaries and human-generated reference summaries in natural language processing. We also include a binary response to represent if a machine-generated summary is on track or not based on human judgment. We believe that having both algorithmic and human approaches allows us to better understand how well a text summarization tool works compared to a human reader.

4.2.1 Rouge

ROUGE is an evaluation metric that is widely used to evaluate text summarization tools. There are various of the ROUGE scoring metric: ROUGE-L, ROUGE-N, ROUGE-W, and ROUGE-S. ROUGE-L measures the longest common sub-sequence (LCS) between both reference and system summaries. ROUGE-N measures the amount of the same N-Grams. ROUGE-S uses skip-bigram based co-occurrence to gain a measurement between the two summaries. Lastly, ROUGE-W is the same as ROUGE-L, but the measurements are weighted. ROUGE-N is the most popular one and it compares the number of the same n-grams (groupings of tokens or words) that show up in the summaries being compared. In this study, we use ROUGE-1 to measure the number of unigrams (single words) because of their simplicity and effectiveness.

4.2.2 On-Track or Off-Track

We also use a binary response to evaluate whether a given summary is on-track or off-track based on the judgment of a human reader. The goal of this additional measurement is to provide a score for the context of the summaries. ROUGE metric measures the similarity between the reference summaries and machine generated summaries. We notice that some machine-generated summaries output sentences with the right key words but either out of context or with a lack of critical information. The On-Track or Off-track metric allows us to rate the summary with either a 1 or a 0 based on a set of established rules. A good summary receives a 1 if it includes the topic in the correct context to the source article or it describes the main subject(s) that the article is covering. Whereas an inadequate summary receive a 0 score if it has poor grammar, completely miss the main topic, or does not summarize the article accurately.

Table 1: Summarization Tools

Tools	Type	Developer	Technique	Description	Training
SMMRY	Ext.	SMMRY[1]	ranking	SMMRY uses a core algorithm to rank sentences with removing transition phrases, unnecessary clauses, and excessive examples. SMMRY's API is sensitive to Unicode characters and it sometimes generate more than one sentence even if the lengh of the output summary is set to one.	none
Sumy	Ext.	Belica [2]	ranking	Sumy uses a ranking sentence algorithm that is sensitive to Unicode characters. This tool has an easy to use API that is accessible in Python.	none
Textteaser	Ext.	Online[3]	ranking	A quick sentence ranking based text summarization tool. Textteaser is sensitive to Unicode characters. Textteaser is developed for python 2 and we updated it for python 3.	none
Bert	Ext.	Google [4]	deep learning	This model is a modified extractive summarizer based on the traditional abstractive Bert model.	pre-trained
Roberta	Abs.	Facebook [5]	deep learning	This tool is a modified Bert model with a better pre-trained model. Its training phase is based on a larger dataset and for a longer amount of time. Reberta removes next prediction objective and dynamically changes the masking pattern applied to the training data.	pre-trained
Bart	Abs.	Facebook [6]	deep learning	Bart is an attempt to make a more powerful model than Roberta. It is a denoising auto encoder for pre-training sequence-to-sequence models. In order for this model to be trained it corrupts text with a noise function and learns by reconstructing the original text. It uses a transformer-based neural machine translation architecture. [3]	pre-trained
DistilBart	Abs.	Hugging Face [7]	deep learning	A faster version of Bart that is supposed to have 97 % of the grammar capabilities of Bart. It is a 40 % reduction of Bart and it runs much faster without GPU acceleration when compared to other abstractive tools. [8]	pre-trained
Pegasus	Abs.	Google [8]	deep learning	This an acronym for Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence models. Pegasus preforms at a similar speed as other heavy abstractive tools used in this study. [10]	pre-trained

4.3 Example: Summarize a News Article

In this section, we use a sample news article published by BBC News on June 16, 2021 [9]. The article covers the story of a father who got PTSD from his daughter's distressing birth and now is encouraging other fathers to talk about their own problems and experience.

Rouge-1	Placeline	First Sent.	Headline	Human	Track
Bart	**0.338**	0.086	**0.153**	0.181	**1**
Bert	0.175	1	0	0.133	0
DistilBart	0.222	0.085	0.049	0.088	0
Pegasus	**0.317**	0.052	**0.129**	0.111	**1**
Roberta	0.222	0.052	0.129	0.166	0
SMMRY	0.229	0.129	0	0.1	0
Sumy	0.175	1	0	0.133	0
Textteaser	**0.317**	0.105	**0.129**	**0.277**	**1**

Table 2: Sample evaluation results of a news article

[9] https://www.bbc.com/news/stories-57442294

Below is a list of machine-generated summaries produced by the select text summarization tools. These summaries were compared to four reference summaries which include the headline, placeline, the first sentence of the original news article and a user-generated summary (Human) using the ROUGE-1 metric.

- SMMRY: "There are stories here and we don't talk about them in public. Most new expectant dads don't know half of what it is like to be a dad, because we don't talk about it." Elliott would like his book to spark new conversations.
- Sumy: Elliott Rae sat on a hot and crowded London Underground Tube and cried without knowing why.
- Textteasor: But in reality everyday life had become a struggle, and the root cause was his daughter's traumatic arrival into the world.
- BART: A father who watched his baby daughter die after she was born with a bacterial infection has spoken out about his experiences of fatherhood and Post Traumatic Stress Disorder (PTSD).
- BERT: Elliott Rae sat on a hot and crowded London Underground Tube and cried without knowing why. "
- ReBERTa: A father whose daughter was born with a ruptured ruptured baby has spoken of the "horrendous" experience of his son's death.
- DistilBart: The A A Whilst Whilst a baby was a bit of a year, the BBC has learned. Time Time, and the UK's most popular - but the time of the year.
- Pegasus: A father who was diagnosed with post-traumatic stress disorder after his daughter was born has written a book about his experiences.

We use four different reference summaries where the first three come from the original article directly and the last one is written by a human reader.

- Placeline: After the distressing birth of his daughter, Elliott Rae struggled with post traumatic stress disorder, but went without help for over a year. He's now urging dads to talk about their problems - and to avoid the agony he went through.
- First Sentence: Elliott Rae sat on a hot and crowded London Underground Tube and cried without knowing why.
- Headline: 'I got PTSD after witnessing my daughter's birth'
- Human Summary: Elliott got PTSD from his daughter's difficult birth and early life health issues.

Table 2 shows the evaluation results when the machine-generated summaries are compared with the reference summaries. In addition, we include a binary response to indicate whether the machine-generated summaries are on track or not. We can observe that BART, Pegasus, and Textteaser consistently outperform other text summarization tools in both the Rouge and On-Track metrics. When the machine-generated summary is compared with the user-generated summary (human), Textteaser achieved a Rouge-1 score of 0.277.

One interesting observation is that BERT and Sumy received the highest possible score of 1 when compared against the first sentence of the original article. However, it is not considered as on track based on human judgment. We noticed that extractive models tend to grab the first sentence as the summary more often than the rest of the article. This creates an unnecessary bias when the first sentence is used as the reference summary. We will discuss this in more detail in section 5.

For this sample article, we rate Pegasus's summary as the best with the most accurate depiction of the subject, timing, and the story line of the original article. Although its Rouge-1 scores are slightly behind Bart, but it includes all the necessary keywords and has the most appropriate context that it is hard for a human reader to distinguish.

Another observation is that BART's summary did include many details such as the abbreviation PTSD and the baby's infection, but unfortunately it also falsely states that the child died. This is an example where the in-dept accurate details outweigh a small inaccuracy, as no other tools were able to pick up such details. The worst summary comes from DistilBart. Its summary has poor grammar and was not able to distinguish that the article was about the father, nor what were the main events.

4.4 Web Application

We made a web application that can take a blob of input text and produce multiple summaries by various text summarization tools in one run. Figure 1 shows the screenshot of the web application that is currently under development. Right now, it only supports Sumy and Textteaser, but it will support all eight summarization tools described in this study by the end of 2021. For each tool, it produces a summary in the Summarized Text section. This application supports both single-thread and multi-threading. Users can also specify the length of the summary to one sentence or more. The application is available online [10].

[2] https://smmry.com/

[3] https://miso-belica.github.io/sumy/

[4] https://github.com/IndigoResearch/textteaser

[5] https://github.com/dmmiller612/bert-extractive-summarizer

[6] https://huggingface.co/transformers/model_doc/roberta.html

[7] https://huggingface.co/transformers/model_doc/bart.html

[8] https://huggingface.co/transformers/model_doc/distilbert.html

[9] https://huggingface.co/transformers/model_doc/pegasus.html

[10] https://tesla.wheatoncollege.edu/summarizers

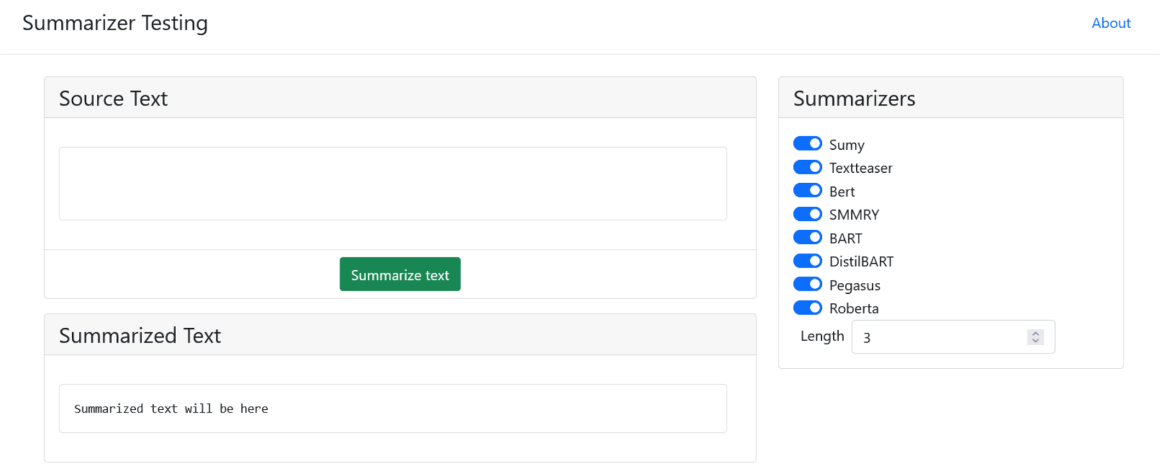

Figure 1: A web application that can take an input text and produce multiple summaries by various text summarization tools in one run. (Work in progress

Rouge-1	Placeline	First Sent.	Headline	Human	On Track
Bart	0.244	0.255	0.177	0.238	61.66%
Bert	0.196	0.965	0.167	0.193	36.66%
distilbart	0.201	0.201	0.130	0.191	5.00%
Pegasus	0.226	0.264	0.171	0.219	40.00%
Roberta	0.210	0.204	0.153	0.233	26.66%
SMMRY	0.185	0.219	0.114	0.202	48.33%
Sumy	0.200	0.481	0.126	0.191	40.00%
TextTeaser	0.162	0.171	0.135	0.184	26.66%

Table 3: Performance evaluation of 8 text summarization tools on 60 news articles

5. EXPERIMENTAL RESULTS

In this section, we present the experimental results of the 8 text summarization tools based on a collection of 60 news articles from various news outlets over the past year. The news articles are fed into our summarization framework to produce summaries and then scored by the evaluation metrics.

Table 3 presents the average Rouge-1 scores when the machine-generated summaries are compared against four reference summaries. From the table, it is clear that two abstractive tools, Bart and Pagusus, are the top two contenders with scores of 0.244 and 0.226 respectively when using the placeline as a reference. When compared with the first sentence, extractive models such as Sumy and Bert tend to choose the first sentence as the output summary more often than others. Bert almost always chose the first sentence resulting in an average score of 0.9646, which was significantly higher than any of the other tools. This makes the first sentence not the ideal reference as it unnecessary bias towards extractive models such as Bert. Thus, we decided not to use the first sentence as reference summary in our future study. When the headline of an article is used as a reference, no text summarization tools stood out of the crowd. Lastly, we compared the machine-generated summaries with a human-written summary. Bart and Pegasus once again outperformed other tools.

In order to provide a different perspective, we also measure the percentage of summaries that are considered on track based on human judgment. The results indicate that Bart leads with a rate of 61.66% meaning that it can produce reasonably accurate summaries for more than three-fifths of the articles. SMMRY is in second and it has an on-track rate of 48.33%. Considering that SMMRY sometimes generates output summaries that have more than one sentence, this gives it an unfair advantage over other tools. Meanwhile, Pegasus, Sumy, and Bart are closely behind with approximately 40% of the on-track rate.

Based on the experimental results of 60 select news articles, we observe that the biggest challenge for the select text summarization tools face is when an article covers multiple storylines. This poses tremendous difficulties as these tools struggle to digest the common theme or the main topic of the article, especially for the extractive tools. Whereas for articles that cover breaking news or a single event, extractive tools can usually generate good summaries quicker than abstractive tools which in general, requires a longer computation time. On the one hand, it is no surprise that extractive tools tend to have more structured summaries with less grammatical errors compared to the abstractive tools. On the other hand, abstractive tools such as Bart and Pegasus do provide more cohesive and accurate summaries but with slightly higher variations and computation time.

6. CONCLUSION & FUTURE WORK

In this survey, we presented a web application that can summarize any given input article by a list of text summarization tools in one run. We also included our preliminary results that show abstractive tools such as Bart and Pegasus and an extractive tool such as SMMRY can produce reasonable accurate summaries. We plan to continue the implementation of the web application and support all text summarization tools with scoring capabilities. We also plan to include more text summarization tools that rely on GPU computing power. Lastly, we would like to conduct a comprehensive study with more text summarization tools and a larger dataset of news articles from various news outlets. We plan to submit our research findings to a conference early next year.

7. REFERENCES

[1] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. Text summarization techniques: A brief survey, 2017.

[2] M. Gambhir and V. Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47:1–66, 2016.

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[4] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*, 2003.

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[6] D. Miller. Leveraging bert for extractive text summarization on lectures, 2019.

[7] N. Moratanch and S. Chitrakala. A survey on extractive text summarization. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pages 1–6, 2017.

[8] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[9] SMMRY. https://https://smmry.com/about. 09 2021.

[10] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2020.