ASSESSING AUTOMATIC EVALUATION METRICS FOR TEXT SUMMARIZATION

ABSTRACT

There are many automatic evaluation metrics available currently for evaluating the machine generated text summaries. But which evaluation metric of all should be chosen? Which one ranks the summaries better? These are the questions that I tried to answer with this project. Few popular metrics have been chosen for this task and had been assessed on how they performed on different summaries an NLP model could generate.

INTRODUCTION TO TEXT SUMMARIZATION

The process of extracting brief and vital information from a large chunk of text data is called as Text Summarization.

There are two main classifications in Text Summarization –

1) Extractive Text Summarization and

2) Abstractive Text Summarization.

In extractive summarization, important sentences from the original text are identified and those exact sentences are extracted and made into a summary.

However, abstractive summarization is a much more advanced method unlike extractive summarization. This method identifies important sentences and understands the context from the source text and generates a summary with new phrases that are still relevant.

For this project we generate an abstractive summary of a news article using the HuggingFace's Transformers.

INTRODUCTION TO EVALUATION METRICS FOR TEXT SUMMARIZATION

Automatic evaluation metrics have been ongoing research since many years. For machine translation**.** They emerged to address the need for objective, consistent, quick, and affordable assessment of MT output.

Some of the metrics assessed in this project are - ROUGE , BLEU, METEOR, and BERTScore.

We will discuss how each of them works and their weaknesses in this article.

CRITERIA FOR ASSESSING THE EVALUATION METRIC

The automatic evaluation metrics for text summarization should be assessed based on the following criteria.

An ideal evaluation metric should rank the summaries based on the below measures -

1. **Relevance**: How relevant is the summary to the source text? Does the summary represent the facts correctly?
2. **Reference clarity:** Are the nouns and pronouns clearly referred to in the summary?
3. **Grammaticality:** Are there any grammatical errors in the summary like incorrect words, incorrect punctuations etc.
4. **Structure and Coherence:** Is the summary well-structured and meaningful?
5. **Semantic Similarity:** How similar are the meanings of the source and summary text?

Now that we understand the criteria, let us discuss how these metrics work.

AUTOMATIC EVALUATION METRICS

1. ROUGE

ROUGE is shorthand for **R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation is a popular automatic evaluation metric. It determines the quality of a summary by comparing it to the summaries created by humans. It measures the number of overlapping units like n-gram, word sequences, and word pairs between machine generated text and the ideal summaries created by the humans.

For this task we use three different ROUGE measures namely, ROUGE-1, ROUGE-2 and ROUGE-L.

1.1. ROUGE-1 and ROUGE-2

ROUGE-1 and ROUGE-2 respectively measure the number of matching 1-grams and 2-grams respectively between a machine generated text and the reference text.

Both can be referred together as ROUGE-N and an n-gram is nothing but a grouping of words or tokens. 1-gram is made up of one word and 2-gram is made up of two consecutive words.

For Example:

Source Text: Sun rises in the east

1-gram: [‘sun’, ‘rises’, ‘in’ , ‘the’, ‘east’]

2-grams: [‘sun rises’,’ rises in’, ‘in the’, ‘the east’]

Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

Where n stands for the length of the n-gram, gramn, and Countmatch(gramn) is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

As it considers total sum of the number of n-grams occurring at the reference summary side in the denominator, it is clearly a recall-based measure. And hence the name.

1.2. ROUGE-L

ROUGE-L measures the longest common subsequence (LCS) between machine summary and reference summary. In other words, we count the longest sequence of tokens that is shared between both. In this method, the summary sentence is treated as sequence of words. The intuition is that the longer the LCS of two summary sentences is, the more similar the two summaries are.

2. BLEU

BLEU, which stands for Bi-Lingual Evaluation Understudy, is another popular automatic evaluation metric. BLEU is a precision-based measure unlike ROUGE which is recall-based method.

BLEU works by computing the n-gram matches sentence by sentence. And next adding the clipped n-gram counts for all the candidate sentences and divide by the number of candidate n-grams. The formula is as follows -

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

Note that, BLEU uses clipped n-gram counts because machine translation systems can over generate "reasonable" words, resulting in improbable, but high-precision, translations.

Moreover, a multiplicative brevity penalty factor is applied to the score to handle machine summarizations that are too short.

$$\text{Brevity Penalty} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

Brevity Penalty (BP) will be 1.0 when the candidate translation length is the same as any reference translation length. The closest reference sentence length is the "best match length."

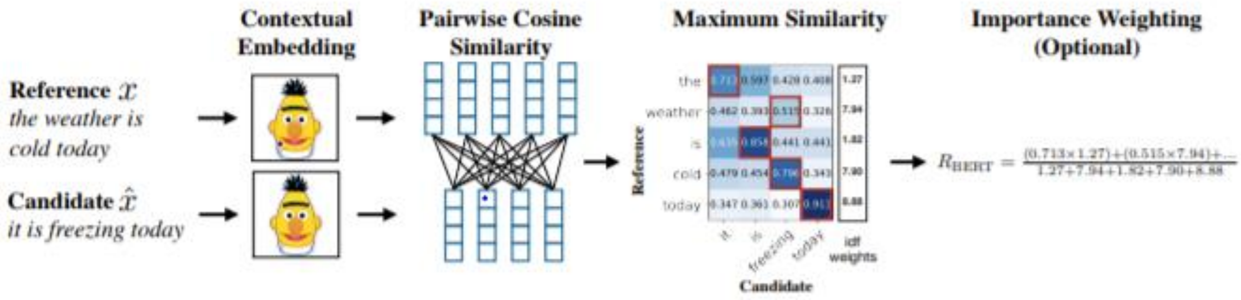
3. METEOR

METEOR stands for **M**etric for **E**valuation of **T**ranslation with **E**xplicit **Or**dering. It is based on a generalized concept of unigram matching between the machine produced summary and human-produced reference summaries. Unigrams can be matched based on their surface forms, stemmed forms, and meanings. Once all generalized unigram matches between the two strings have been found, METEOR computes a score for this matching using a combination of unigram-precision, unigram-recall, and a measure of fragmentation that is designed to directly capture how well-ordered the matched words in the machine translation are in relation to the reference.

4. BERTScore

All the three metrics discussed above – ROUGE, BLEU, METEOR is all n-gram based metrics. Where as BERTScore is an embedding-based metric. BERTScore computes a similarity score for each token in the candidate sentence with each token in the reference sentence. However, instead of exact matches, we compute token similarity using contextual embeddings.

Given the reference x and candidate xˆ, we compute BERT embeddings and pairwise cosine similarity. Computation of recall metric for BERTScore is illustrated below.



The complete score matches each token in x to a token in xˆ to compute recall, and each token in xˆ to a token in x to compute precision. Greedy matching is used to maximize the matching similarity score, where each token is matched to the most similar token in the other sentence.

The recall, precision and F1 scores are given as:

$$R_{\mathrm{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^{\top} \hat{\mathbf{x}}_j \ , \quad P_{\mathrm{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^{\top} \hat{\mathbf{x}}_j \ , \quad F_{\mathrm{BERT}} = 2 \frac{P_{\mathrm{BERT}} \cdot R_{\mathrm{BERT}}}{P_{\mathrm{BERT}} + R_{\mathrm{BERT}}} \ .$$

So, now that we understand how these metrics work, lets get our hand dirty with coding to assess these metrics.

CODE IMPLEMENTATION

- **Installing Dependencies and Importing Libraries**

```
!pip install sentencepiece bert-score transformers datasets rouge_score
!pip install -U nltk
from transformers import pipeline
import pandas as pd
from datasets import load_metric
```

- **Loading the popular CNN/DailyMail Dataset**

```
from google.colab import drive
drive.mount('/gdrive',force_remount=True)
import zipfile
with zipfile.ZipFile('/gdrive/MyDrive/CNN DailyMail Dataset/archive (6).zip',
'r') as zip_ref:
    zip_ref.extractall('gdrive/MyDrive/')
dataset = pd.read_csv("/content/gdrive/MyDrive/cnn_dailymail/test.csv")
dataset.head(1)
```

- **Summarizing a news article**

I am using Huggingface's Transformer's pipeline method to perform summarization task using a BART model trained on CNN articles. Then generating the summary for the selected article.

BART is a denoising autoencoder for pretraining sequence-to-sequence models. It uses encoder of BERT and decoder of GPT.

```
summarizer = pipeline("summarization", model="facebook/bart-large-cnn")
summary = summarizer(dataset.article[8458])[0]['summary_text']
```

- We get the human generated reference and BART generated summary as below

Human generated summary/ Golden Reference:

Dan Price, founder of Gravity Payments, will slash his salary to $70,000 and raise the minimum wage for his 120 employees to $70,000 .

The move, in the next three years, will also require Price to plow up to 80percent of his firm's $2.2million profits back into salaries .

He says he was inspired to make the bold step after reading that happiness increases dramatically for people earning over $70,000 .

Machine generated summary/ Candidate:

Gravity Payments CEO Dan Price is slashing his $1million salary to the same as his lowest-paid worker. He's also taking up to 80percent of the firm's $2.2million expected profit to plow it back in to staff salaries. Over the next three years, Gravity Payments will offer at least $70,000 salary to all of its 120 employees - even the most junior customer service representatives and clerks.

- **Evaluating the BART summary**

We now evaluate the summary produced by the BART model and use those metrics to compare the metrics of other manipulated summaries to evaluate how the metrics perform.

Loading the metrics that we are going to assess from huggingface's datasets library.

```
from datasets import load_metric
rouge_metric = load_metric('rouge')
bleu_metric = load_metric('bleu')
meteor_metric = load_metric('meteor')
bertscore_metric = load_metric('bertscore')
```

Computing ROUGE

```
results_rouge=rouge_metric.compute(predictions=[summary],references=[dataset.highlights[8458]])
```

R-1 Score:

precision=0.5714285714285714, recall=0.547945205479452, fmeasure=0.5594405594405595

R-2 Score:

precision=0.2318840579710145, recall=0.2222222222222222, fmeasure=0.22695035460992907

R-L Score:

precision=0.3, recall=0.2876712328767123, fmeasure=0.2937062937062937

Computing BLEU

```
results_bleu = bleu_metric.compute(predictions=[summary.split()],references=[[dataset.highlights[8458].split()]])
```

'bleu': 0.10225026454683218

Computing METEOR

```
results_meteor = meteor_metric.compute(predictions=[[summary]],references=[[dataset.highlights[8458]]])
```

'meteor': 0.4904703619513901

COMPUTING BERTScore F1 measure

```
results_bertscore = bertscore_metric.compute(predictions=[summary],references=[dataset.highlights[8458]],lang='en')
```

'f1': [0.9044607877731323]

- **Assessing the metrics**

I will manipulate the BART generated summary and generate 5 new summaries on which I will compute the chosen evaluation metrics against the human generated reference and see how they react to assess them. We manipulate the summaries by - Synonymizing of words, re-ordering the words, changing the facts, introducing grammatical mistakes, and paraphrasing.

Metrics after replacing few words in the BART summary with their synonyms

I substituted the following words with their respective synonyms -

slashing -> cutting

plow it back -> reinvest it

offer -> provide

firm -> companies

employees -> workers

salary -> compensation

Therefore, we have the below synonymized summary -

"Gravity Payments CEO Dan Price is cutting his $1million compensation to the same as his lowestpaid worker. He's also taking up to 80percent of the coompany's $2.2million expected profit to reinvest it in to staff compensation. Over the next three years, Gravity Payments will provide at least $70,000 compensation to all of its 120 workers even the most junior customer service representatives and clerks ”

Now we compute all the metrics using this summary with original article summary as the reference.

```
syn_results_rouge=rouge_metric.compute(predictions=[syn_summary],references=[dataset.highlights[8458]])
syn_results_bleu = bleu_metric.compute(predictions=[syn_summary.split()],references=[[dataset.highlights[8458].split()]])
syn_results_meteor = meteor_metric.compute(predictions=[[syn_summary]],references=[[dataset.highlights[8458]]])
syn_results_bertscore = bertscore_metric.compute(predictions=[syn_summary],references=[dataset.highlights[8458]],lang='en')

print("Rouge score after swapping words with its synonyms")
print(syn_results_rouge)
print("BLEU score after swapping words with its synonyms")
print(syn_results_bleu)
print("METEOR score after swapping words with its synonyms")
print(syn_results_meteor)
print("BertScore after swapping words with its synonyms")
print(syn_results_bertscore)
```

We get the below scores –

‘rouge1’:

precision=0.4927536231884058, recall=0.4657534246575342, fmeasure=0.4788732394366197

‘rouge2’:

precision=0.17647058823529413, recall=0.16666666666666666, fmeasure=0.17142857142857143

‘rougeL’:

precision=0.2608695652173913, recall=0.2465753424657534, fmeasure=0.25352112676056343

'bleu': 0.08617000865817602

'meteor': 0.4040326719623007

bertscore - 'f1': [0.8922862410545349]

Then we make another manipulated summary and compute the metrics. I am not pasting the code as the code is similar as above.

Metrics after randomly disordering the words in BART summary

I randomly made a new summary by disordering the words. The summary then becomes meaningless but the words and the number of words that the summary contain is the same.

So, this is the disordered summary –

“Gravity Payments is CEO Dan Price slashing his $1million as his lowest paid worker salary to the same. He's also taking up to it back expected profit 80percent of the firm's $2.2million to plow in to staff salaries. Over the next three years, Gravity Payments will offer at least $70,000 salary to all of its 120 employees junior the most customer service representatives even and clerks.”

After computing the metrics these are the scores –

‘rouge1’:

precision=0.5714285714285714, recall=0.547945205479452, fmeasure=0.5594405594405595

‘rouge2’:

precision=0.21739130434782608, recall=0.20833333333333334, fmeasure=0.2127659574468085

‘rougeL’:

precision=0.2714285714285714, recall=0.2602739726027397, fmeasure=0.26573426573426573

'bleu': 0.08617000865817602

'meteor': 0.4040326719623007

bertscore -'f1': [0.8922862410545349]

Metrics after changing the facts of the BART summary

I produced another summary by distorting the facts. We changed all the numbers and also replaced some words with their antonyms. Ex: I changed the words -

all -> none

junior -> senior

taking -> not taking

slashing -> not slashing

So the new summary is,

“Gravity Payments CEO Dan Price is not slashing his $100million salary to the same as his lowest-paid worker. He's also not taking up to 60percent of the firm's $100.2million expected profit to plow it back in to staff salaries. Over the next three years, Gravity Payments will offer at least $20,000 salary to none of its 12000 employees - even the most senior customer service representatives and clerks.”

After computing the metrics these are the scores –

‘rouge1’:

precision=0.5, recall=0.4931506849315068, fmeasure=0.496551724137931

‘rouge2’:

precision=0.14084507042253522, recall=0.1388888888888889, fmeasure=0.13986013986013987

‘rougeL’:

precision=0.25, recall=0.2465753424657534, fmeasure=0.2482758620689655

'bleu': 0.06161188792969887

'meteor': 0.41683753419826974

bertscore -'f1': [0.8881691098213196]

<u>Metrics after introducing grammatical mistakes to the BART summary</u>

I made another summary by introducing grammatical mistakes. We did so by the changing the capital letters to small and changing the words –

is -> are

to -> for

his -> her

taking -> took

to -> too

it -> them

at least -> atleast

Below is the new summary,

“Gravity Payments CEO dan price are slashing his $1million salary for the same as her lowest-paid worker.he's also took up too 80percent of the firm's $2.2million expected profit to plow them back in to staff salaries.over the next three years, Gravity Payments will offer atleast $70,000 salary to all of its 120 employees - even the most junior customer service representatives and clerks.”

After computing the metrics these are the scores –

‘rouge1’:

precision=0.5507246376811594, recall=0.5205479452054794, fmeasure=0.5352112676056338

‘rouge2’:

precision=0.20588235294117646, recall=0.19444444444444445, fmeasure=0.19999999999999998

‘rougeL’:

precision=0.2753623188405797, recall=0.2602739726027397, fmeasure=0.2676056338028169

'bleu': 0.06665525302235266

'meteor': 0.47629627903673927

bertscore -'f1': [0.8823244571685791]

Metrics after paraphrasing the BART summary

I produced a semantically and factually correct, paraphrased summary and evaluated it.

"CEO of Gravaity payments,Dan Price, is taking a pay cut to match the pay rate of his employees. He is also re-investing the comapany's 80percent of profit into staff salaries. For the next three years, Gravity Installments will offer at slightest $70,000 compensation to all of its 120 workers - including the most junior customer service representatives and clerks. He was motivated after reading about how increase in the salary raises the happiness levels in people"

After computing the metrics these are the scores –

‘rouge1’:

precision=0.46835443037974683, recall=0.5068493150684932, fmeasure=0.4868421052631579

‘rouge2’:

precision=0.1282051282051282, recall=0.1388888888888889, fmeasure=0.13333333333333333

‘rougeL’:

precision=0.25316455696202533, recall=0.273972602739726, fmeasure=0.2631578947368421

'bleu': 0.05973297672718817

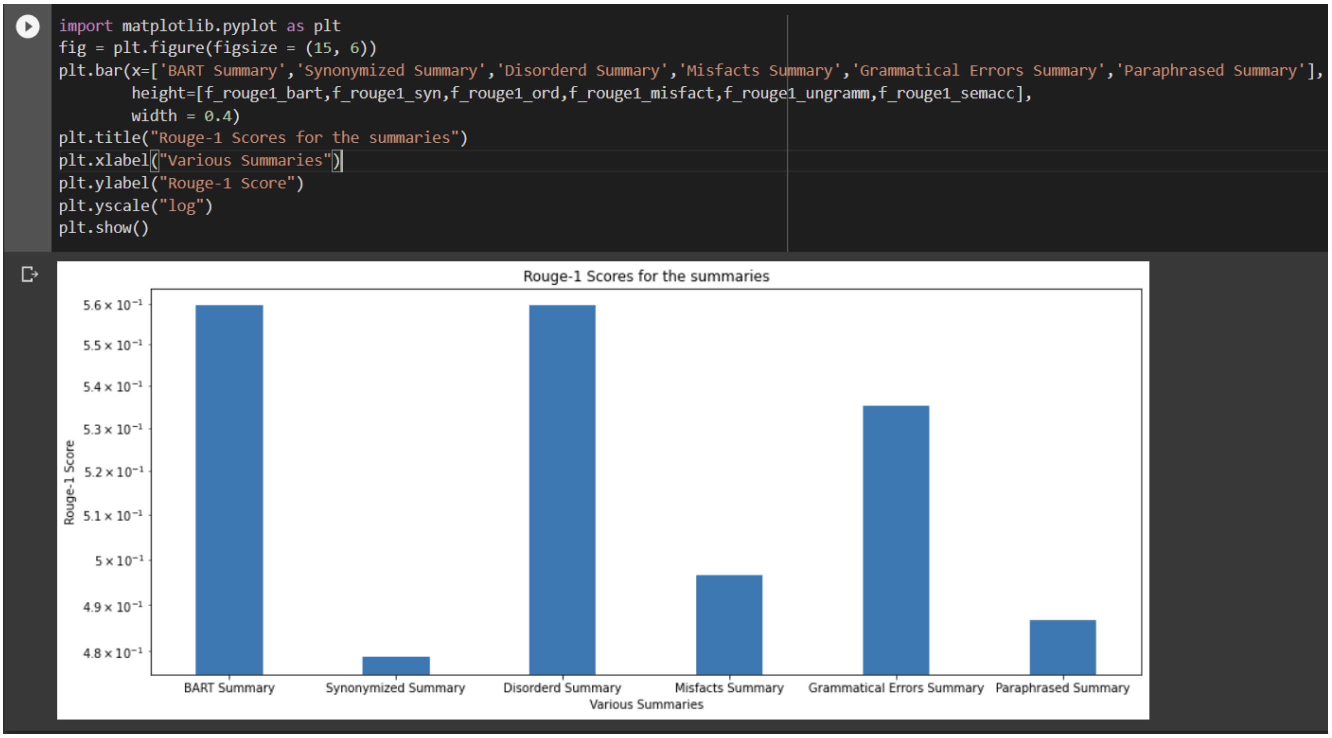
'meteor': 0.40694953153079

bertscore -'f1': [0.8845058679580688]

I am only considering the F1 score from each metric to compare and asses them.

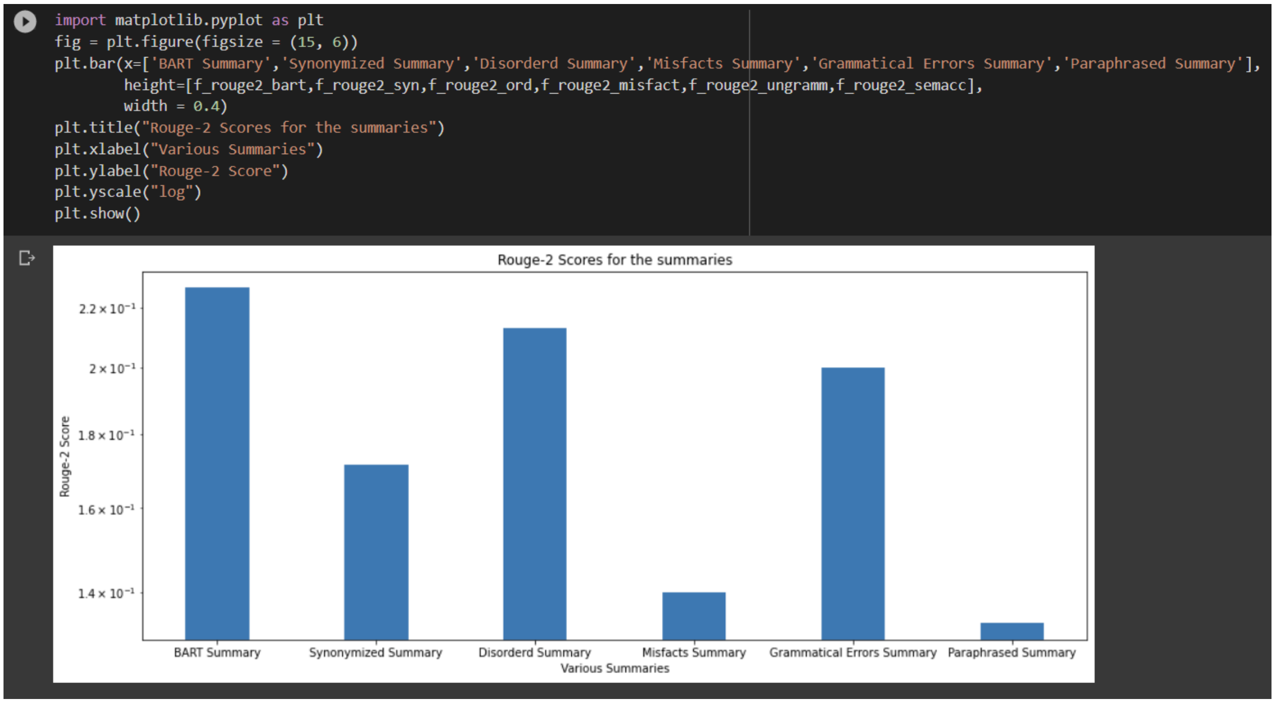
- **Plotting the metrics**

Comparing R-1 scores for all the summaries



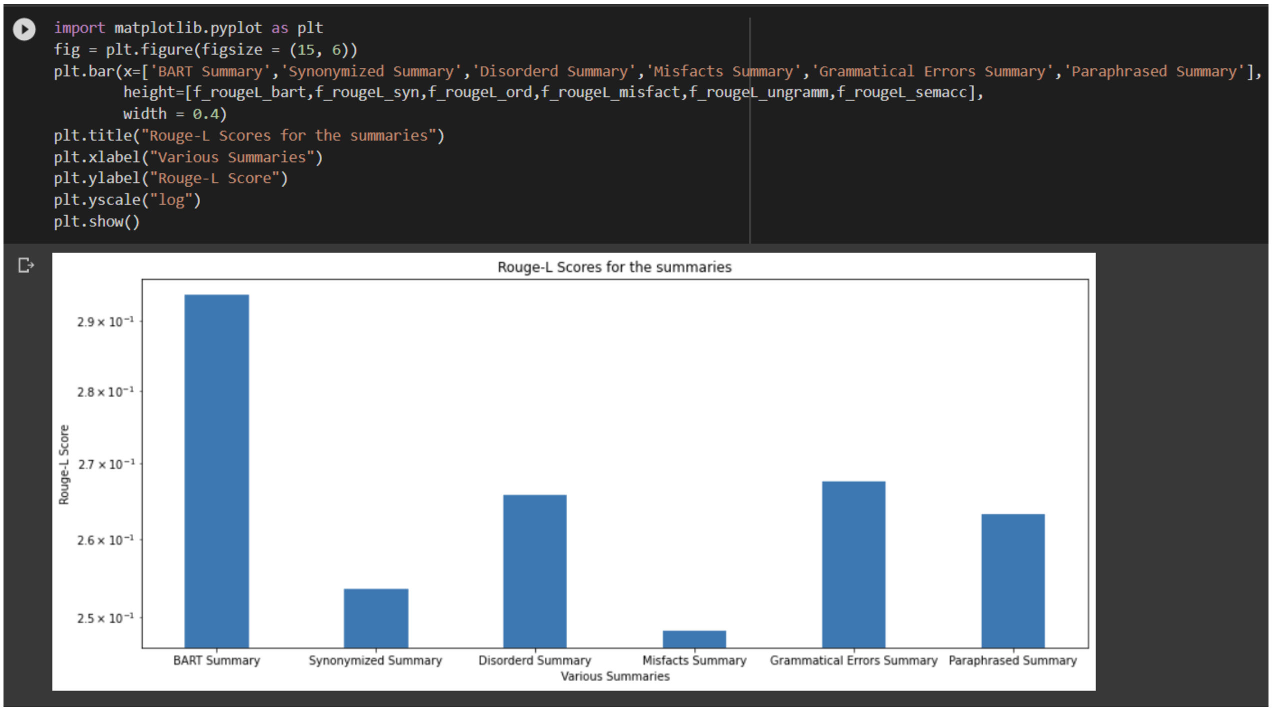
```
import matplotlib.pyplot as plt
fig = plt.figure(figsize = (15, 6))
plt.bar(x=['BART Summary','Synonymized Summary','Disorderd Summary','Misfacts Summary','Grammatical Errors Summary','Paraphrased Summary'],
        height=[f_rouge1_bart,f_rouge1_syn,f_rouge1_ord,f_rouge1_misfact,f_rouge1_ungramm,f_rouge1_semacc],
        width = 0.4)
plt.title("Rouge-1 Scores for the summaries")
plt.xlabel("Various Summaries")
plt.ylabel("Rouge-1 Score")
plt.yscale("log")
plt.show()
```

Comparing R-2 scores for all the summaries



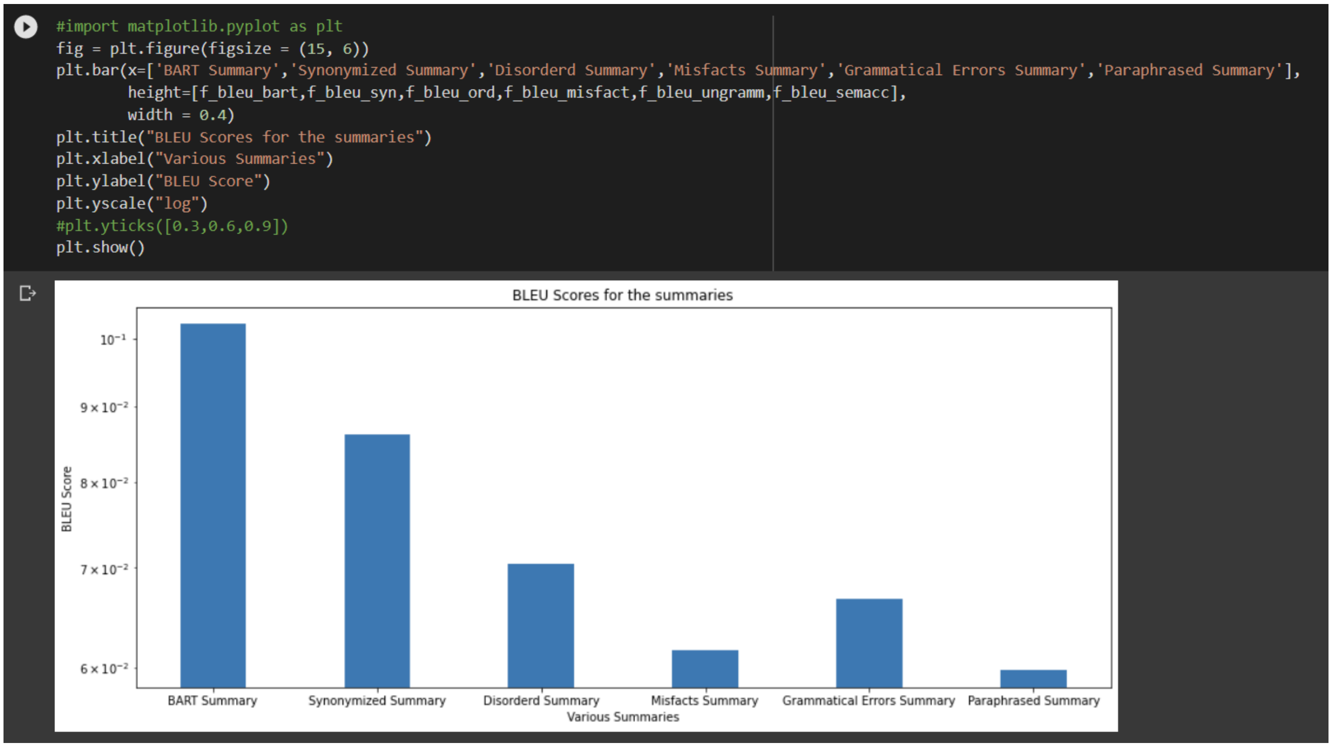
```
import matplotlib.pyplot as plt
fig = plt.figure(figsize = (15, 6))
plt.bar(x=['BART Summary','Synonymized Summary','Disorderd Summary','Misfacts Summary','Grammatical Errors Summary','Paraphrased Summary'],
        height=[f_rouge2_bart,f_rouge2_syn,f_rouge2_ord,f_rouge2_misfact,f_rouge2_ungramm,f_rouge2_semacc],
        width = 0.4)
plt.title("Rouge-2 Scores for the summaries")
plt.xlabel("Various Summaries")
plt.ylabel("Rouge-2 Score")
plt.yscale("log")
plt.show()
```

Comparing R-L scores for all the summaries



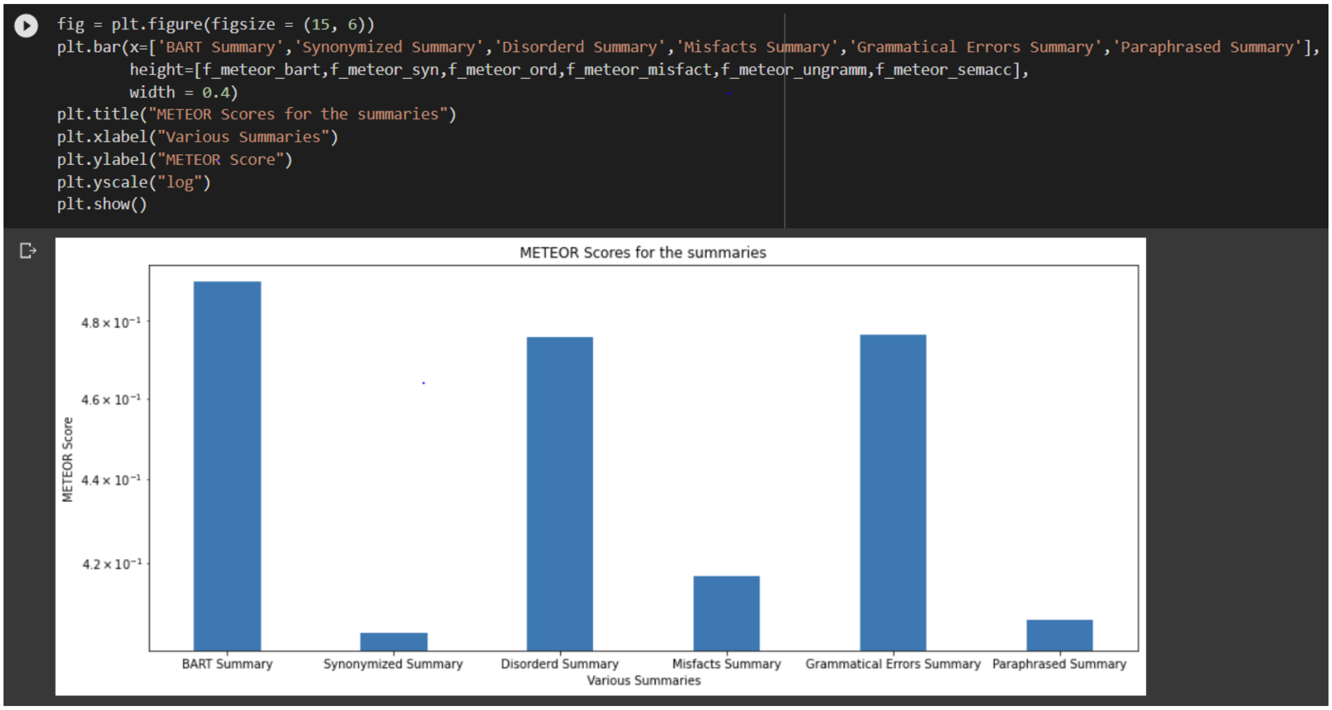
```
import matplotlib.pyplot as plt
fig = plt.figure(figsize = (15, 6))
plt.bar(x=['BART Summary','Synonymized Summary','Disorderd Summary','Misfacts Summary','Grammatical Errors Summary','Paraphrased Summary'],
        height=[f_rougeL_bart,f_rougeL_syn,f_rougeL_ord,f_rougeL_misfact,f_rougeL_ungramm,f_rougeL_semacc],
        width = 0.4)
plt.title("Rouge-L Scores for the summaries")
plt.xlabel("Various Summaries")
plt.ylabel("Rouge-L Score")
plt.yscale("log")
plt.show()
```

Comparing BLEU scores for all the summaries



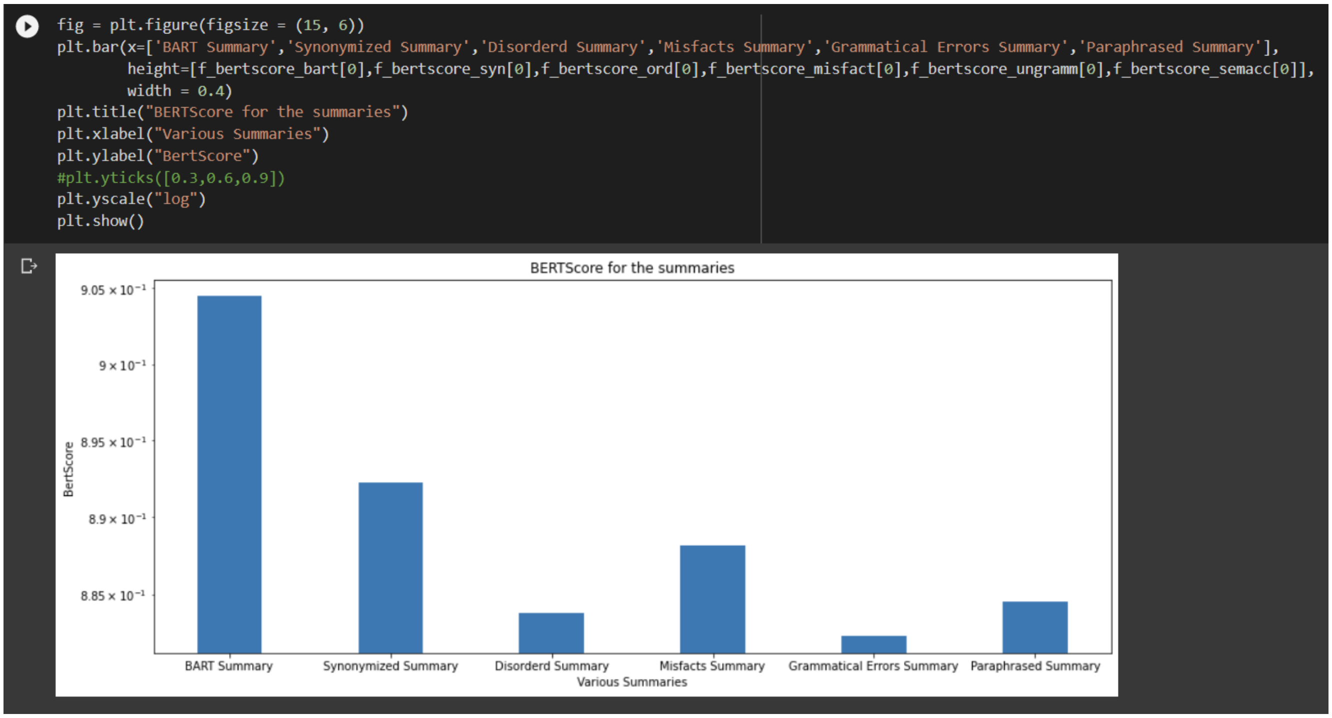
```
#import matplotlib.pyplot as plt
fig = plt.figure(figsize = (15, 6))
plt.bar(x=['BART Summary','Synonymized Summary','Disorderd Summary','Misfacts Summary','Grammatical Errors Summary','Paraphrased Summary'],
        height=[f_bleu_bart,f_bleu_syn,f_bleu_ord,f_bleu_misfact,f_bleu_ungramm,f_bleu_semacc],
        width = 0.4)
plt.title("BLEU Scores for the summaries")
plt.xlabel("Various Summaries")
plt.ylabel("BLEU Score")
plt.yscale("log")
#plt.yticks([0.3,0.6,0.9])
plt.show()
```

Comparing METEOR scores for all the summaries



```
fig = plt.figure(figsize = (15, 6))
plt.bar(x=['BART Summary','Synonymized Summary','Disorderd Summary','Misfacts Summary','Grammatical Errors Summary','Paraphrased Summary'],
        height=[f_meteor_bart,f_meteor_syn,f_meteor_ord,f_meteor_misfact,f_meteor_ungramm,f_meteor_semacc],
        width = 0.4)
plt.title("METEOR Scores for the summaries")
plt.xlabel("Various Summaries")
plt.ylabel("METEOR Score")
plt.yscale("log")
plt.show()
```

Comparing BERTScore scores for all the summaries



```
fig = plt.figure(figsize = (15, 6))
plt.bar(x=['BART Summary','Synonymized Summary','Disorderd Summary','Misfacts Summary','Grammatical Errors Summary','Paraphrased Summary'],
        height=[f_bertscore_bart[0],f_bertscore_syn[0],f_bertscore_ord[0],f_bertscore_misfact[0],f_bertscore_ungramm[0],f_bertscore_semacc[0]],
        width = 0.4)
plt.title("BERTScore for the summaries")
plt.xlabel("Various Summaries")
plt.ylabel("BertScore")
#plt.yticks([0.3,0.6,0.9])
plt.yscale("log")
plt.show()
```

- **Inference from the plots**

ROUGE:

- ROUGE-1, ROUGE-2 scores for the meaningless disordered summary were almost same as that of the BART generated summary. In other words, according to R1 and R2, the disordered summary is as good as the meaningful and coherent BART generated summary, which is clearly not.
- All three Rouge scores gave good score for summary with grammatical errors.
- Synonymized and Paraphrased summaries have performed worst according to the rouge scores. Which are better summaries.

Therefore, Rouge doesn’t perform well according to our criteria.

BLEU:

- BLEU scored the synonymized summary better than other manipulated summaries. But a little worse than the BART generated summary, which shouldn't have been the case.
- BLEU noticeably scored the summary with mis-facts and synonymized summary well.
- But the BLEU couldn't judge the paraphrased summary.

Therefore, BLEU is not an ideal metric.

METEOR:

- METEOR rated disordered and grammatical error summaries high.
- It didn't rate the synonymized and paraphrased summary well.

BERTScore:

- Clearly scored the disordered and grammatical error summaries correctly.
- Score for the synonymized summary is not bad as well.
- But could not score the paraphrased summary properly.

CONCLUSION

From the inference made from the above graphs, I conclude that - among the all the metrics that we evaluated, BERTScore should be the outright choice to evaluate a machine generated text summary.

REFERENCES

https://openreview.net/pdf?id=SkeHuCVFDr

https://www.cs.cmu.edu/~alavie/papers/BanerjeeLavie2005-final.pdf

https://arxiv.org/pdf/2008.12009.pdf

https://aclanthology.org/P02-1040.pdf

https://aclanthology.org/W04-1013.pdf

https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460

Code Implementation for this article can be found at:

Assessing Automatic Evaluation Metrics For Text Summarization.ipynb - Colaboratory (google.com)