**EVALUATING TEXT SUMMARIZERS**

**Evaluation using ROUGE Metrics**

ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a metric that is used to obtain the degree of similarity between a candidate summary (automatically generated summary) and the target summary (hand-written reference summary. ROUGE scores are divided into ROUGE-1, ROUGE-2, and ROUGE-L scores.

**ROUGE-1**

ROUGE-1 compares the degree of similarity of *unigrams* in the automatically generated and hand-written summaries. Unigrams, in this case, are words. Thus, the precision and recall can be calculated by evaluating how many individual words are captured from the source material to the final automatically generated summary.

If we have a candidate sentence: "He went to the park"

This sentence can be expressed as a list of unigram tokens:  
'He','went','to','the','park'

Let us say we have a reference sentence: "He went to the park yesterday"

This sentence contains the following unigram tokens:  
'He','went','to','the','park','yesterday'

Now, we look at all of the unigram tokens captured by the candidate sentence:  
'He','went','to','the','park'

Thus, the ROUGE-1 Precision can be calculated as:  
(Number of captured unigram tokens) ÷ (Number of candidate unigram tokens)

This gives us 5 ÷ 5 = 1

And, the ROUGE-1 Recall can be calculated as:  
(Number of captured unigram tokens) ÷ (Number of reference unigram tokens)

This gives us 5 ÷ 6 = 0.83

The ROUGE-1 F-Score can be calculated as:  
2 x (Precision x Recall) ÷ (Precision + Recall)

This gives us 2 x (1 x 0.83) ÷ (1 + 0.83) = 0.907

#### ROUGE-2

ROUGE-2 compares the degree of similarity of bigrams in the automatically generated and hand-written summaries. Bigrams, in this case, are two consecutive words. Thus, the precision and recall can be calculated by evaluating how many bigrams are captured from the source material to the final automatically generated summary.

If we have a candidate sentence: "He likes going to the park"

This sentence can be expressed as a list of bigram tokens:  
'He likes','likes going','going to','to the','the park'

If we have a reference sentence: "He really likes going to the park"

This sentence contains the following bigram tokens:  
'He really','really likes','likes going','going to','to the','the park'

Now, we look at all of the bigram tokens captured by the candidate sentence:  
'likes going','going to','to the','the park'

Thus, the ROUGE-2 Precision can be calculated as:  
(Number of captured bigram tokens) ÷ (Number of candidate bigram tokens)

This gives us 4 ÷ 5 = 0.8

And, the ROUGE-2 Recall can be calculated as:  
(Number of captured bigram tokens) ÷ (Number of reference bigram tokens)

This gives us 4 ÷ 6 = 0.66

The ROUGE-2 F-Score can be calculated as:  
2 x (Precision x Recall) ÷ (Precision + Recall)

This gives us 2 x (0.8 x 0.66) ÷ (0.8 + 0.66) = 0.723

#### ROUGE-L

ROUGE-L measures the longest common subsequence (LCS). This refers to the words that happen to be in sequence, not taking into account any different words that are in the way of the matching sequence (when comparing the candidate and reference sentences).

For example, if we have the candidate sentence: "I carried an umbrella to the zoo in case it rained"

This sentence contains the following tokens:  
'I','carried','an','umbrella','to','the','zoo','in','case','it','rained'

If we have a reference sentence: "I took an umbrella to the zoo since it could have rained"

This sentence contains the following tokens:  
'I','took','an','umbrella','to','the','zoo','since','it','could','have','rained'

Now, we look at all of the captured tokens:  
'I','an','umbrella','to','the','zoo','it','rained'

Thus, the ROUGE-L Precision can be calculated as:  
(Number of captured tokens) ÷ (Number of candidate tokens)

This gives us 8 ÷ 11 = 0.72

And, the ROUGE-L Recall can be calculated as:  
(Number of captured tokens) ÷ (Number of reference tokens)

This gives us 8 ÷ 12 = 0.66

The ROUGE-L F-Score can be calculated as:  
2 x (Precision x Recall) ÷ (Precision + Recall)

This gives us 2 x (0.72 x 0.66) ÷ (0.72 + 0.66) = 0.688

## List of Text Summarization Models

Now that we know about the two broad categories of summarization models, as well as the evaluation metrics that we will use to score our automatically generated summaries, let us take a look at the different models that we will be comparing in this article.

1. Luhn's Heuristic Method
2. TextRank
3. Latent Semantic Analysis (LSA)
4. Kullback-Leibler Sum (KL-Sum)
5. T5 Transformer Model

#### Luhn's Heuristic Method

Luhn's Heuristic Method for Text Summarization is one of the first Text Summarization algorithms, being published in 1958. It is based on TF-IDF (Term Frequency-Inverse Document Frequency), and selects words of high importance based on their frequency of occurrence. Also, higher weightage is given to the words that occur at the beginning of the document.

#### TextRank

TextRank is a graph-based extractive Text Summarization technique. It is used to find the most relevant sentences (as well as keywords) in a piece of text. Here, sentences that contain highly frequent words are considered important. Thus, the algorithm assigns scores to each sentence in the source material. The sentences are then ranked in the descending order of their scores, and the top scoring sentences are included in the summary.

#### Latent Semantic Analysis (LSA)

Latent Semantic Analysis is an unsupervised Natural Language Processing (NLP) technique that uses statistics to extract the association between words in a document on the basis of their contextual use. The goal is to identify the most important topics from the source material and to then choose the sentences with the greatest combined weights across the topics. Singular Value Decomposition (SVD) is the statistical technique that is used to uncover the hidden semantic structure of words in the source material. Let us now generate an automatic summary using Latent Semantic Analysis.

#### Kullback-Leibler Sum

In the realm of mathematical statistics, the Kullback-Leibler divergence, which is also often termed relative entropy, is a type of statistical distance, that is used to measure how different a probability distribution 'P' is when compared with a reference probability distribution 'Q'. It is inversely proportional to the degree of similarity between the source material and the automatically generated summary (in terms of readability and the information conveyed). The Kullback-Leibler Sum algorithm is a greedy method that creates a summary by appending sentences as long as the Kullback-Leibler divergence is decreasing. This ensures that the summary contains a set of sentences that happen to be similar to the document set unigram distribution.

#### T5 Transformer Model

Transformers are a type of neural network architecture, and were developed by a group of researchers at Google (and UoT) in 2017. They avoid using the principle of recurrence, and work entirely on an attention mechanism to draw global dependencies between the input and the output. Transformers allow for much more parallelization than sequential models, and can achieve very high translation quality even after being trained only for short periods of time. They can also be trained on very large amounts of data without as much difficulty. Read more about 'Text Summarization using Transformers' [here.](http://https/iq.opengenus.org/text-summarization-using-transformers/)

The T5 Transformer model (developed by Google AI in 2020) is an encoder-decoder model that can achieve state-of-the-art results when performing Natural Language Processing (NLP) tasks, while also being flexible enough to be fine-tuned for more specific problems. It frames all such tasks to a text-to-text format, where the input and output are always strings.