Comparative Analysis of NLP models for Google Meet Transcript Summarization

Yash Agrawal[1], Atul Thakre[2] , Tejas Tapas[3], Ayush Kedia[4], Yash Telkhade[5], Vasundhara Rathod[6]
[1]Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, India

[2]Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, India
[3]Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Indi
[4]Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, India
[5]Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, India
[6]Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, India
****Author for Correspondence*** E-mail: agrawaly_2@rknec.edu, Tel: 7083645470, E-mail: thakrear@rknec.edu, Tel: 8956758226

Abstract
Manual transcription and summarization is a cumbersome process necessitating the development of an efficient automatic text summarization technique. In this study, a Chrome extension is used for making the process of transcription hassle free. It uses the text summarization technique to generate concise and succinct matter. Also, the tool is accessorized using Google Translation, to convert the processed text into users' desired language. This paper illustrates, how captions can be traced from the online meetings, corresponding to which, meeting transcript is sent to the backend where it is summarized using an NLP model. It also walks through three different NLP models and presents a comparative study among them. The NLTK model utilizes the sentence ranking technique for extractive summarization. Word Embedding model uses pre -trained Glove Embeddings for extractive summarization. The T5 model performs abstractive summarization using transformer architecture. The working of the model is tested over meeting texts taken from various sources and results show that the NLTK model has an edge over the Word Embedding model based on ROUGE-1, ROUGE-2, and ROUGE-L scores. However, our analysis finds that T5 is generating a more concise summary.

Keywords: *text summarization, precision f measure, automatic text summarization, word embedding model, f measure recall, measure recall precision, natural language processing, automatically generated summary, abstractive summarization, vector representation, extractive summarization, park reference summary, child went home, human generated summary, model summary, chrome extension, human generated summary rouge , sentence score, extractive text summarization model, meet transcript*

INTRODUCTION

In today's world, an enormous amount of textual material is generated and is only growing every single day. On average 2.5 quintillions, bytes of data are produced by humans every day. In such a scenario manually analyzing and interpreting text becomes difficult. A large amount of infrastructure and human resources are wasted in processing a huge amount of data. Consuming the data in its original unstructured form is time-consuming and inefficient. This paper talks about data generated from online video conferencing platforms like google meet. Statistics show that there is an increase in 87% of people using video conferencing systems for daily communication purposes. Also, $37 billion is wasted annually in the U.S. on unproductive meetings, hence arising the need for automatic text summarization. Automatic Text Summarization is an AI-driven process that comprises of making a concise text using the most important information such that its meaning is not changed. This helps in reducing manual effort and is much required in online meetings. It also helps in generating summaries that are not biased in comparison to human-generated summaries. This Paper uses Natural Language Processing(NLP) Model as a tool for automatic text summarization. NLP, a sub-field of Artificial Intelligence enables a computer to read, hear, interpret a text in human language and can also determine which parts of the text are im]portant. Hence NLP is the most appropriate tool to produce a summary.

TYPES OF TEXT SUMMARIZATION

Automatic Text Summarization is mainly divided into 2 categories, extractive, and abstractive summarization based on the nature of generated summarized text. Extractive summarization performs the summarization of text by extracting the most important sentences of the entire text. The subset of text appears as it is in the summarized text. It is a selection-based approach. Abstractive Summarization attempts to simulate the human ability to generate entirely new sentences without misrepresenting the meaning of the original text. This paper presents a comparative analysis of the above two models.

METHODOLOGY USED

DATA PREPROCESSING

The received text from the chrome extension consists of redundant information like the name of the speaker, date, and other unnecessary bits of information. This data needs to be cleaned, integrated, transformed, and loaded before it passes through the NLP model for summarization. Hence data pre-processing holds great importance before the ultra-processing step ushers in.

STOP WORDS REMOVAL

During text pre-processing, we generally come across some words that are trivial in the text and can be removed. Removing such words helps to reduce the storage space in the database and saves a lot of pre-processing time which is necessary to extract important words from the text. Examples of stop words in English are “a”, “the”, “is”, “are”, etc. which can be removed from our text in order to make our processed content more efficient as they do not contribute to any future operations.

TOKENIZATION

Tokenization is a key step in every summarization technique, whether it is abstractive or extractive, either it is using the bag of words technique or advanced neural network architecture. There are 3 types of tokenization: sentence tokenization, word tokenization, and character tokenization. Unlike English, some languages use punctuation marks as a part of a word and, whitespaces cannot be used as a delimiter of words. Word tokenization is specifically used for creating vocabulary in a particular context. The next step that follows tokenization is to represent each token in mathematical language. For instance, some encoding uses simple numbering to represent words, while complex word embeddings use multidimensional vector representations of the word. These representations can then be used for finding the semantic relationship between different tokens. Similar to word tokenization, sentence tokenization is useful in finding similarities between different sentences and also to establish the importance of a particular sentence in the given text. The process of sentence selection is the basis of many extractive summarization techniques.

STEMMING AND LEMMATIZATION

Stemming and Lemmatization are widely used for text preprocessing. The basic aim of Stemming and Lemmatization is to generate the root word of the inflected words. Stemming is the process of reducing inflected words to their word stem. It is the process that just finds the base word which may or may not have a meaning. Lemmatization on the other hand will always give a meaningful word representation. For words like finally, final, finalized Stemming will generate its root word as fina, which does not have any meaning in the English language. While Lemmatization will generate the final as a root word that is meaningful.

VARIOUS MACHINE LEARNING MODEL

NLTK MODEL

Natural Language Toolkit (NLTK) is a python library that is a powerful tool for Computational Linguistics. It is a text processing library for human language data and is fabled for supporting powerful functions like text tokenization, stemming, text classification, etc. It is an extractive text summarization model and works on the principle of generating summaries by selecting top-ranked sentences.

Example: Let the input text be: "A child goes to the park. The child starts playing in the park. In the evening, the child went home."

It goes through given five processes:

Data Cleaning

Text: child goes park. Child starts playing park. Evening, child went home.

Tokenization

Word Tokenization: ['child', 'goes', 'park', '.', 'child', 'starts', 'playing', 'park', '.', 'evening', ',', 'child', 'went', 'home','.']

Sentence Tokenization: ['child goes park.', 'child starts playing park.', 'evening, child went home.']

Generating Word frequency table

This step involves finding the frequency for each word in the text [refer Table 1]. This step becomes necessary as it will be used in determining sentence scores.

TABLE 1

Word	Frequency
child	3
goes	1
park	2
starts	1
plain	1
evening	1
went	1
home	1

Sentence Scoring

Here sentence score is calculated and allocated to each sentence present in the text based on the summation of word frequency for each word present in that sentence

TABLE 2

Sentence	Sentence Score
child goes park	3 + 1 + 2 = 6
child starts playing park	3 + 1 + 1 + 2 = 7
evening child went home	1 + 3 + 1 + 1 = 6

Summary

Sentences are prioritized according to their sentence scores and a summary is selected by selecting the top ranking sentences based on a predefined compression ratio that can be specified by the user.

Word Embedding Model

Glove stands for Global Vector Representation, which is a pre-trained dataset [4] of word embeddings. It is trained on 6 Billion words with a vocabulary size of 400,000. Each word is represented as a 100 Dimensional Vector. This vector is obtained by using an unsupervised machine learning algorithm operated on aggregated global word-word co- occurrence statistics. The limitation with models which are based on bag of words approach or TF-IDF (Term Frequency – Inverse Document Frequency) approach is that they use simple incremental encoding which does not convey any semantic relationship or similarity between words.

Consider 4 words "King", "Queen", "Man", "Woman". Now let V-King be the 100 D vector representation for the word “King” and similarly for other words.
Then, V-Queen = V-King – V-man + V-Woman.
This example clearly shows the correlation between vectors of words that are closely related to each other. Thus the use of these vectors for the representation of tokens can prove to be quite effective.

Just like the Bag of words model, the first step will be to get some kind of one hot encoding representation, which is meant to create a vocabulary of the given text. The glove vectors can then be used to create an embedding matrix for the vocabulary. This is the base for the representation of the corpus that is to be summarized. A consolidated vector representation for tokenized sentences can be created using the embedding matrix representation.
To build a semantic relationship between different sentences of the corpus, some kind of mathematical measure is needed for calculating the similarity between each pair of sentences. This paper uses cosine similarity as a measure for calculating the similarity between vector representations.

If X and Y are 2 vectors of the form (X1, X2, X3……….X100) and (Y1, Y2, Y3 Y100), then cosine similarity between the 2 vectors can be calculated as:

$$\text{Cosine similarity (X, Y)} = \frac{X.Y}{|X| \times |Y|}$$

where |X| is magnitude of the vector X and can be calculated as:

$$|X| = \sqrt{X.X}$$

This similarity matrix is converted into a graph, with sentences as nodes and similarity values as edges between those nodes. This representation can then be passed to an algorithm for calculating the scores of sentences in the graph. This paper uses the TextRank algorithm for score calculation. Sentences are then ranked on basis of scores and top N sentences can be picked. The value of N can be selected based on a specific use case and the amount of compression required for the original text.

T5 Model

The Transformer intends to resolve sequence-to-sequence tasks to handle the dependencies between input/output with recurrence and attention. T5 [5], text-to-text-transfer-transformer plays a significant role under abstractive summarization giving effective results. The capabilities of transfer learning come from pre-training a model on largely available scripts with self-monitoring tasks, such as language modelling, or by placing in the lacking words.

Eventually, the model can be calibrated on smaller labelled datasets for achieving greater throughput on the labelled script itself.

In recent times, XLNet, Reformer, BERT are the fabled methods of transfer learning, and thus the presence of an enormous amount of these methods has caused a perplexing situation to predict which amongst them is the most high- performing one. Unlike BERT-style models, T5 reframes all Natural Language Processing tasks into a text-to-text format in which the input/output is always in the form of text strings.

T5 performs four main tasks : Summarization, Translation, Classification, and Regression

The chrome extension has made use of the Summarization module of T5 which works in 4 steps

Mapping the words of the text to unique identifiers.

Those unique identifiers are then mapped to the vectors from training and representation of words which is known as encoding

Model generation using 'T5ForConditionalGeneration' with 't5-small' as a pre-trained dataset.

Finally decoding those to the human-readable summarized text.

The various other uses of T5 include - closed book questionnaires and even fill in the blanks and one-word questions. The amount of Flexibility provided by T5 makes it versatile and adaptive to many different applications.

ROUGE ANALYSIS

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a metric for evaluating summaries generated by NLP models. It counts the number of overlapping units in a human-generated summary and automatically generated summary from NLP models. It is used to measure Precision, Recall, and F measure.

Precision: It measures how much the automatically-generated summary was relevant or needed.

$$Precesion = \frac{No\ of\ overlapping\ words}{No\ of\ words\ in\ refrence\ summary}$$

Recall: It measures how much of the human-generated summary the automatically-generated summary is resembling.

$$Recall = \frac{No\ of\ overlapping\ words}{No\ of\ words\ in\ system\ summary}$$

F measure: It is the harmonic mean of Precision and Recall.

$$F\ measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

There are different methods to calculate the overlap between summaries [6]:
ROUGE - 1: It measures the presence of individual words (UNIGRAM) of automatically-generated summary in human-generated summary
ROUGE - 2: It detects the presence of BIGRAM (pair of adjacent words) of automatically-generated summary in human-generated summary
ROUGE - L: It calculates overlap by using Longest Common Subsequence Algorithm. It does not require predefined n-gram length.
Example:
Model Summary: Rahul was found near the park Reference Summary: Rahul was near the park
Model Summary Unigrams: Rahul, was, found, near, the, park Reference Summary Unigrams: Rahul, was, near, the, park

ROUGE 1 (Precision): $\frac{5}{6}$ ROUGE 1 (Recall): $\frac{5}{5}$

Model Summary Bigrams: Rahul was, was found, found near, near the, the park
Reference Summary Bigrams: Rahul was, was near, near the, the park

ROUGE 2 (Precision): $\frac{3}{6}$ ROUGE 2 (Recall): $\frac{3}{5}$

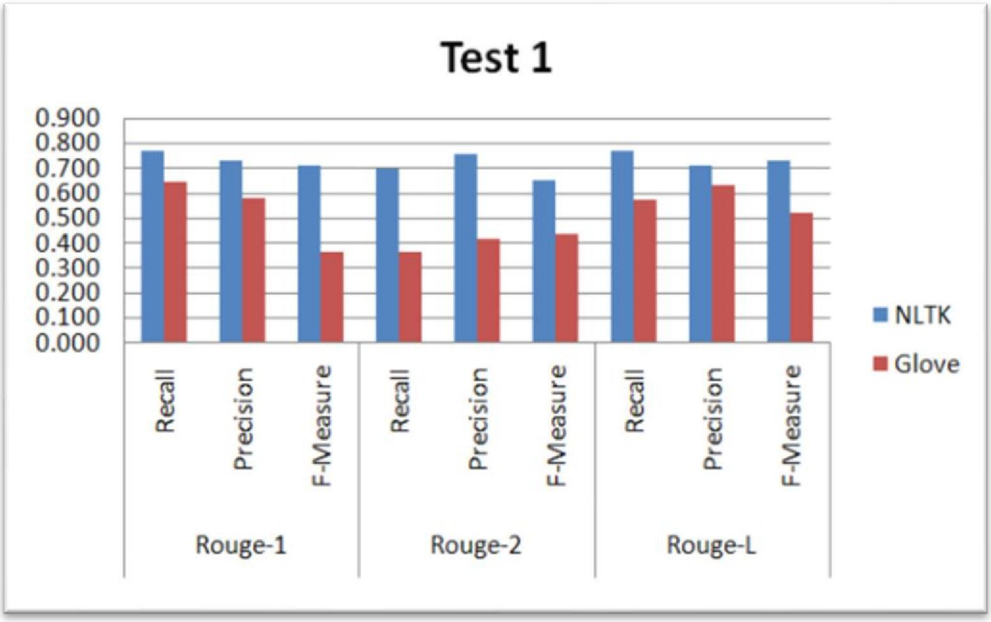
ROUGE L (Precision): $\frac{3}{6}$ ROUGE L (Recall): $\frac{3}{5}$

For comparison of Extractive Models i.e. NLTK and Word Embedding, BBC News Summary Dataset [7] (Dataset exclusively for Extractive text summarization models and contains around 417 political news. It has one model summary and other is reference summary) was used.

Both models were used to generate a summary and, ROUGE-1, ROUGE-2, and ROUGE-L values were calculated for each summary. Fig 1 shows comparative analysis of different ROUGE values for a particular text in the dataset.
It is evident from Fig 1 that ROUGE-1 and ROUGE-L values are greater than the ROUGE-2 values. A more or less similar trend was observed for other articles as well. It is observed that NLTK model performed better in ROUGE Analysis than the Word Embedding model.

TEST 1 ANALYSIS

Model		Rouge-1			Rouge-2			Rouge-L	
	Recall	Precision	F-Measure	Recall	Precision	F-Measure	Recall	Precision	F-Measure
NLTK	0.767	0.731	0.713	0.701	0.759	0.651	0.769	0.710	0.732
Glove	0.646	0.579	0.363	0.367	0.417	0.435	0.573	0.635	0.521

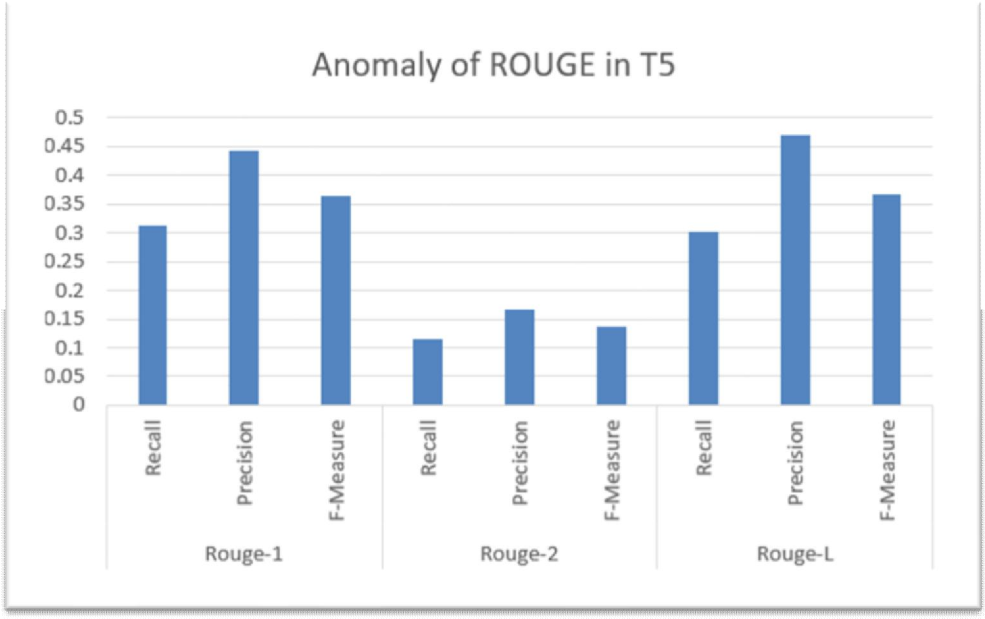


Anomaly of Rouge for T5

T5 is based on the abstractive summarization which introduces new words, hence the use of the overlapping word as a metric is not appropriate. Till now no concrete parameter for evaluation of abstractive summary exists, however, some preliminary research is going on in the field of ROUGE-AR [8] , where AR stands for Anaphora Resolution.

TEST 2 ANALYSIS

Model	Rouge-1			Rouge-2			Rouge-L		
	Recall	Precision	F-Measure	Recall	Precision	F-Measure	Recall	Precision	F-Measure



CONCLUSION

The NLP models presented in this paper were tested over 100 google meetups with different topics and intents in every meeting. The meet transcript was generated with the relevant information like hostname, meeting Id, and date of the meetup. After data pre-processing, the key points from the meet transcripts were successfully extracted and converted to human-readable text.

The paper covered three models for text summarization. The two of which, NLTK and Glove model were the extractive summarizers and T5 was an abstractive text summarizer. After comparing the results generated by these summarizers, it was found that T5 was giving the best results which were suitable for the chrome extension.

Hence the model successfully extracted the meet transcripts and converted them to the human-readable text using abstractive summarizer - T5.

The future scope of this project is to expand it further to other video conferencing tools such as Zoom, Microsoft Teams, Cisco WebEx, Skype, and BlueJeans meetings.

REFERENCES

1. Deepali K. Gaikwad and C. Namrata Mahender, A Review Paper on Text Summarization (International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016)
2. Divya Khyani, Siddhartha, Niveditha N M,Divya B M, An Interpretation of Lemmatization and Stemming in Natural Language Processing (Journal of University of Shanghai for Science and Technology)
3. Giorgio Maria Di Nunzio and Federica Vezzani, A Study on Stemming vs Lemmatization (A Linguistic Failure Analysis of Classification of Medical Publications)
4. Jeffrey Pennington, Richard Socher, Christopher D. Manning, GloVe: Global Vectors for Word Representation (Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543,October 25-29, 2014)
5. Adam Roberts, Staff Software Engineer and Colin Raffel, Exploring Transfer Learning with T5: the Text-To- Text Transfer Transformer
6. Chin-Yew Lin, ROUGE: A Package for Automatic Evaluation of Summaries (In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004))
7. Pariza Sharif, BBC News Summary Extractive Summarization of BBC News Articles (Source Kaggle)