Live Summarization (Streaming Data)

Mohd. Zartab Ali
zartab19041@iiitd.ac.in
Rohit Ghai
rohit19105@iiitd.ac.in
Dhawal Singh Pundir
dhawal19120@iiitd.ac.in

1 Abstract

The automatic summarization is a novel technique that involves the use of statistical methods to extract out the main context and meaning from a given piece of text information. The methods of summarization are largely being used in various organizations to present a generic report out of unstructured and bulky datasets. There have been various research conducted in order to efficiently summarize the raw text information. The studies conducted so far involves a varied set of techniques that uses mathematical and probabilistic methods to get the best content out of the text. This paper proposes sentence ranking algorithms based on graph and feature word extraction to generate the summary.

2 Introduction

Document Summarization or Text Summarization is the process of creating a concise summary of the document preserving its important information and its meaning. Summarization reduces the text by eliminating the less useful information that helps the reader to quickly find required information. There are two ways to do the above task:-

- Abstractive Summarization
- Extractive Summarization

Abstractive Summarization methods generate sentences based on semantic representation. It uses natural language generation techniques to produce a summary which is closer to what human readers generate while reading a document (1). Extractive Summarization methods work by selecting a subset of words, sentences and phrases from the original document to generate a summary.

In order to tackle the problem of text summarization we have used mainly Extractive Summarization (1) methods along with a certain degree of abstractive approach. In the end we have performed comparative analysis of the approaches used on the basis of summary evaluation metrics.

3 Problem Definition

In today's world, a lot of textual data is generated via various platforms such as social media, blogs, news articles, web pages, tweets etc. This data is increasing day by day. Most of this data is unstructured and it takes a lot of time and human effort to navigate through the data to get useful results. There is a need to reduce this type of data to a shorter version so that readers can easily and quickly extract the useful information from that, and this can be achieved by text summarization. We are extending this problem of text summarization to summarize live streaming data such as data from twitter and other blogging sites.

4 Background

It is not feasible for every reader to read the whole content from a source as it is time taking and requires a lot of human effort and concentration. So there is a need for text summarization. Text summarization refers to the technique of shortening long pieces of text. When a person reads a text, he develop the understanding about the content and then write the summary which contains the main points of that text. Since computers lack human intelligence and semantic knowledge, it makes automatic text summarization a challenge for the computers.

5 Dataset Used

Initially, we have used a dataset of News Articles. It has been scraped from https://www.thenews.com.pk website. It consists of news articles from 2015 related to business and sports.The tables consist of 4 columns namely Article, Date, Heading and NewsType. The article content also

contains the place where the article was published. It contains 2693 rows. This dataset is available on Kaggle: https://www.kaggle.com/asad1m9a9h6mood/news-articles/download

Further, we used real time twitter data which is collected using tweepy library. Tweepy is a Python library(API) used for accessing Twitter. We have used 20 different categories. Some of them are Article370, Boyslockerroom, ChattisgarhGasLeak, Coronavirus etc. Currently, each of our documents consist of 20 tweets. The count is dynamic and we can increase or decrease the number of tweets in a document.

In order to preprocess the tweets, we removed the username, the url, tags etc. which are present in the tweets. Then sentence tokenization is applied and sentences are extracted from the tweets. These sentences together comprise a document.

6 Baselines

So as a baseline model, we developed the tf-idf vector and cosine similarity methods to rank sentences of a particular document in its summary. To start with, tweets are combined to form a document, pre-processed and sentences are extracted. Vectorization is applied and we have maintained a dictionary for every word which contains idf value. A vector of [1 X N] is maintained for every sentence which contains tf-idf value for each word in the sentence, here N is the size of the vocabulary (2).

Cosine score is calculated for a sentence with respect to all sentences in that document. Each score is normalized using the length of the sentences.

$sentencescore(i) = \text{for each j}\Sigma(cosine - similarity(score_{ij}))/N$

Where N is the total number of sentences in the document.

By doing this, longer sentences will not get more preference over the shorter one. After sorting all the sentences with respect to cosine score top root N sentences will be output by the system where N is the number of total sentences in a document. Root N document is taken to maintain the size of summary proportional to document size.

Here is the short visualization shown in Fig. 1. It shows some of the words along with the frequency which is indicated by word size. It helps us to detect the frequently occurring words in the document.

Figure 1: WordCloud

7 Evaluation Criteria

To evaluate baseline & other proposed models, we used ROUGE - Recall-Oriented Understudy for Gisting Evaluation (3). It is specifically used for evaluating automatic summarization of texts. It compares automatically produced summary and a set of reference summaries (produced by humans) i.e. ground truth. For generating the ground truth, we manually summarize some of the documents (4).

In ROUGE, recall is defined as how much of reference summary the system summary is recovering or capturing. So, considering individual words,
Recall = Number of Overlapping Words / Total Words in Reference Summary

In ROUGE, precision is defined as how much of system summary was in fact relevant or needed? So, considering individual words,
Precision = Number of Overlapping Words / Total Words in System Summary

And then we calculate, F-Measure. So, by combining these 3 types of ROUGE scores are calculated. These are ROUGE-N, ROUGE-L & ROUGE-S.

- **ROUGE - 1 (R1)** : It measures unigram overlap ratio between the texts.
- **ROUGE - 2 (R2)** : It measures the bigram overlap ratio between the texts.
- **ROUGE - L (RL)** : It measures the longest matching subsequence of words using LCS.

8 Proposed Algorithms

8.1 TextRank Algorithm

In this, we created a graph of the document as shown in 2. Each sentence of a document is represented by a node and the cosine similarity value between the two sentences is used to represent an edge between a pair of nodes (5). Then the well known Page Rank algorithm that is used to compute ranking of the nodes in the graph based on the structure of the incoming links. It was originally designed as an algorithm to rank web pages. Alpha is a damping parameter for PageRank, default value is 0.85. The nodes are retrieved in decreasing order of their scores.

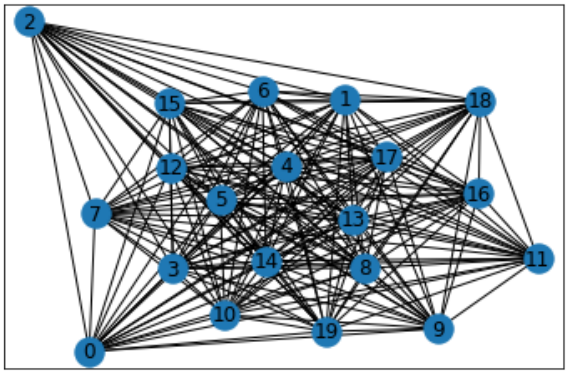

Figure 2: Graph

$Rankofsentence(u) = (1 - d)/N + d\Sigma_{\text{for each sentence v}}(Rank(v)/degree(v))$

Here 'd' is the damping parameter which determines the probability of teleport operation.This model is evaluated using ROUGE Evaluation Matrix and we get output as:
R1 = 0.429970194256755
R2 = 0.31464774062763856
R-L = 0.4729847659811698
Precision = 0.3416217476218165
Recall = 0.6111300763447731

The R1 score for various algorithms is shown in Fig. 3

8.2 Hybrid Similarity

In this we combined Graph Ranking, POS Tagging, Feature Extraction using Idf and Synsets (6). The sentences are tokenized into words and words are ranked according to their Idf values. The top 10 words are given extra weightage alpha as they are important in summarization. Further, POS Tagging

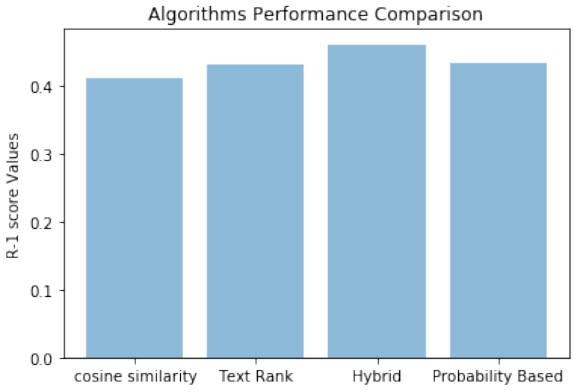

Figure 3: R1 Score

(7) of sentences is done and a word is given weightage beta, if it is a noun, noun phrase, an adjective etc. Then the Synset is created i.e groupings of synonymous words that express the same concept that is used to capture the semantic similarity among tweets. It captures the similarity between the word pairs in the hypernym tree of the words. The nltk wordnet module is used for pos tagging and synsets generation. In order to calculate the similarity between synsets Wu-Palmer similarity between the sets is calculated.

$Wu - Palmersimilarity(S1, S2) = depth(LCS(S1, S2))/(depth(S1) + depth(S2))$

Here S1, S2 are the synsets for the word1 and word2. Mathematically score for each sentence is given as,

score-sentence (i) = Initial-Score (i) + α * Initial Score + (β) * $_{\text{for each word j in sentence}}$ Initial Score + $_{\text{for each word j in sentence}}$ wup-similairty * Initial Score + γ * $_{\text{for-topical word in sentence}}$ Initial Score

Here, Initial Score - is the text-rank score of sentence(i).
α (alpha) - factor multiplied for sentences containing more than 10 words.
β (beta) - factor multiplied if the word in a sentence is "Noun or Adjective".
γ (gamma) - factor multiplied if the word in a sentence is topical word.

The sentences are then used to summarize the document. This model is evaluated using ROUGE Evaluation Matrix and we get output as:
R1 = 0.4599398822033337
R2 = 0.3724618351814811
R-L = 0.4928960602426785
Precision = 0.3303452542110713
Recall = 0.7746178232016776

The R2 score for various algorithms is shown in Fig. 4

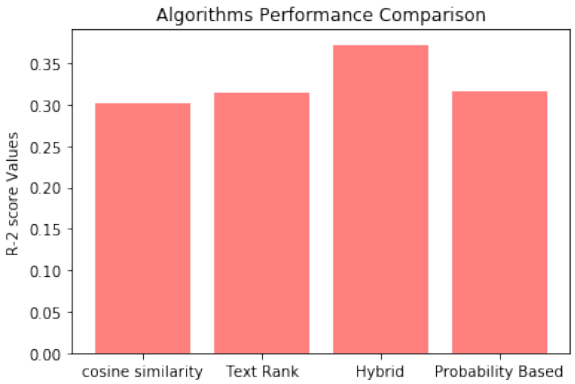

Figure 4: R2 Score

8.3 Probability Based

In this method, top words are extracted using their Idf values. Then the probability of occurrence of each and every word is calculated in a document (8). For the feature extraction of the topic words idf for each word is calculated considering each sentence as document.The both Idf values and probability values are combined for every word. Then the word scoring is added up to generate sentence scoring and then sentences having higher values are picked in summary of a document.

Score-Sentence(i) = $\sum_{\text{for each for "w" in the document}}$ P(w) + idf(w)

Here, idf is the inverse-document frequency of the word and "P" is the probability of occurrence of the word in the document.

This model is evaluated using ROUGE Evaluation Matrix and we get output as:
R1 = 0.43296196762226147
R2 = 0.316655308439052
R-L = 0.4459821719554739
Precision = 0.3936091765305796
Recall = 0.5007971524459409

The RL score for various algorithms is shown in Fig. 5

9 Literature Review

There are many research papers related to text summarization. In these research papers authors have described many techniques to handle the text summarization problem. Some of these techniques are Intermediate Representation, Sentence Score, Summary Sentences Selection, Frequency-driven Approaches, Latent Semantic Analysis, Bayesian

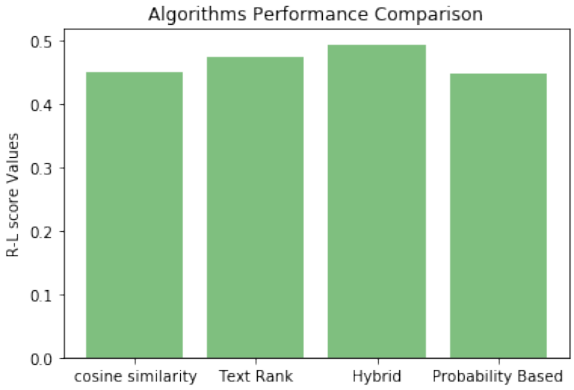

Figure 5: RL Score

Table 1: ROUGE Score of Different Algorithm's

Algorithm	**R-1**	**R-2**	**R-L**
Cosine Similarity	0.411	0.30	0.45
Textrank	0.429	0.314	0.472
Hybrid	**0.46**	**0.37**	**0.493**
Probability	0.432	0.316	0.445

Topic Models, Machine Learning for Summarization, etc (9). As most of the work is done in summarization on static data and live summarization of text is rare. So we have worked on live text summarization of data from various social media platforms such as twitter.

10 Results Produced

As discussed above, we have used ROUGE Evaluation Matrix, the results produced for different methods are shown in Table 1.

The Precison and Recall values for different algorithms is shown in Table 2 .

The Precison comparison using bar graph is shown in Fig. 6 and Recall comparision is shown in Fig. 7

Table 2: Precicion Recall for Different Algorithm's

Algorithm	**Precision**	**Recall**
Cosine Similarity	0.336	0.583
Textrank	0.34	0.61
Hybrid	0.33	**0.773**
Probability	0.39	0.50

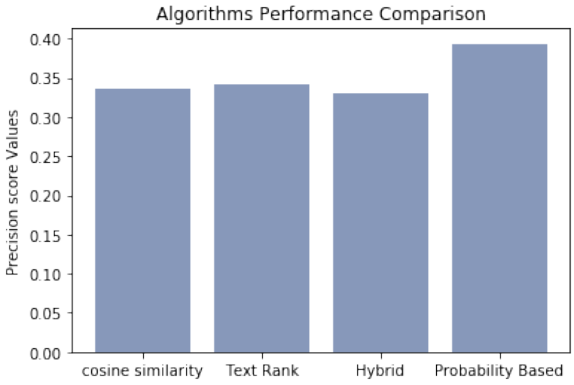

Figure 6: Precision

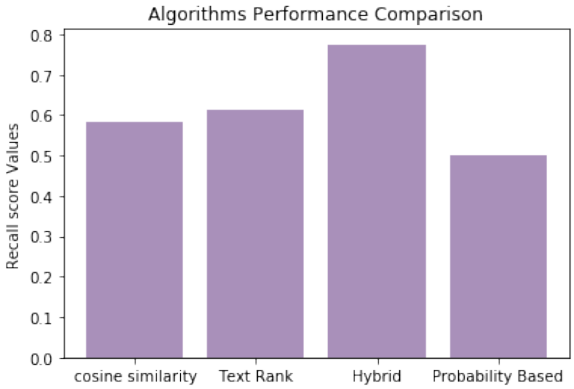

Figure 7: Recall

References

[1] https://ieeexplore.ieee.org/abstract/document/8817040

[2] https://ieeexplore.ieee.org/document/8989281

[3] https://medium.com/free-code-camp/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840

[4] https://www.aclweb.org/anthology/W04-1013/

[5] https://ieeexplore.ieee.org/document/6909126

[6] https://ieeexplore.ieee.org/document/8991312

[7] https://ieeexplore.ieee.org/document/7096234

[8] https://arxiv.org/pdf/1707.02268.pdf

[9] https://ieeexplore.ieee.org/document/7562805