International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

Machine Learning Approach on Paragraph Summarization

Mr. Aniruddha K V[1], Mr. Anup Kumar N Joshi[1], Mr. Lokesh S [2]

[1]Student,Dept. Of Computer Science Engineering, The National Institute Of Engineering, Mysuru, India
[2]Associate Prof.,Dept. Of Computer Science Engineering, The National Institute Of Engineering, Mysuru, India

---

Abstract – *In this growing world, Data is being generated at a greater speed.To summarize a given data manually is practically not feasible. So achieve a goal of obtaining summarized data we use a method called summarization. Summarization process classified as an abstractive and extractive method, where abstractive creates a Summary by understanding the meaning and analysing the document. Extractive creates a summary by extracting sentence which contains maximum information.*

Key Words: *Paragraph summarization, Tokenize , Ranking , Frequency , Voting Model.*

1. INTRODUCTION

The world wide web provided us with huge amount data which is getting increased beyond the limit. In fact every second the amount of data which is getting generated is a lot. To analyze, the given data manually through the intent,it is almost an impossible task, as part of improving quality of data the paragraph summarization came into existence. The attempt has started long back but in the recent year the process is growing efficiently with help of new technology.

The paragraph summarization has been taken from branch called Machine learning , where the machine is trained in order to predict and to provide the future data by using previous data. Paragraph summarization involves in between steps to obtain the result , they are training to rank the sentences, classifying the sentences using priority and provides the final summary. The program basically uses some part of Natural Language Processing for ranking sentences.

2. RELATED RESEARCH

Paragraph Summarization is being used in many field in order to obtain the efficient Data content from a text document. By Dharmendra hingu and Deep shah explains that text is first preprocessed to tokenize the sentences and performs operation. Yogesh kumar and Meena explains optimal features set for extractive automatic text summarization. Wjogan , jaya kumar and ong singh explains abstract voting model for summarized extraction from text document .

3. IMPLEMENTATION

A mentioned above, there are 2 different methods of Implementing the process of paragraph summarization . Here we are implementing by using Extractive method in which sentences are being extracted based on rating the words. Sentence in the given paragraph. After this process the data is being predicated by the machine in order to provide the required summary to the uses. There the user can also obtain the required summary to the user . There the user can also obtain the required amount of summary by specifying the percentage output . This helps in providing an efficient data summary for the input data. The Summarization system is as shown below in Fig 1.

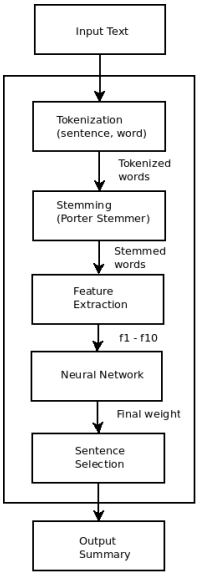

Fig 1 : Paragraph Summarization System

3.1 TF - IDF Algorithm

Typically, the Tf-Idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

- **TF: Term Frequency**

  It measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

  TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

- **IDF: Inverse Document Frequency**,

  It measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones , by computing the following :

  IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

See below for a simple example.

Example:

Consider a document containing 100 words wherein the word *cat* appears 3 times. The term frequency (i.e., tf) for *cat* is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word *cat* appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

3.2 Word frequency:

In any document there will many important terms that will appear frequently in given document. This is behind the word frequency which we use TF IDF.

3.3 Title word:

The frequency of word in a sentence from document title shows the importance of sentence which is highly related to document.

F_0(s)=(Number of Title words in S/Number of words in the title)

3.4 Sentence positioning:

In a paragraph, introductory and conclusive sentences are important. The sentence will be given move weight than remaining sentence.

F_1(s)=(assigned position value/total importance)

3.5 Sentence similarity:

This explain how sentence linked with other sentences. This is calculated by using common words between 2 sentences and dividing it with length of longer sentences. This is also called Bushy path method. Here graph is created as an edge weight and sentences as Nodes. If it bound out b below the threshold value, the edges are removed. The number of edges are weight if the sentence. The sentences have very little relevance with each other. Similarity with remain sentence in the paragraph. This explains how sentences are related to enter paragraph. The word in in sentence which actually matches with other words in the document are counted and linked by total words in the document.

F_2(s)=(Keyword in s and key in other side/ key word in the other sentence)

3.6 Cue word:

This indicates sentences which hold important information in the document.

Eg: (“significantly”);

3.7 Named Entities :

Usually the sentences containing named entities will be having key information. Hence the weight more for that sentences.

$F_3(s)$=number of named entities in s

(or)

$F_3(s)$=(key word in s and keyword in title/keyword in title)

3.8 Length sentence:

Usually long sentence contain more information than short one. Some Short sentence contain no information. This feature measure sentence length which is very important. Fig 2 show the length of sentences and total number of words obtained from the given sample input for the summarization process.

*$F_4(s)$=(Analyze sentence length * length of (s))*

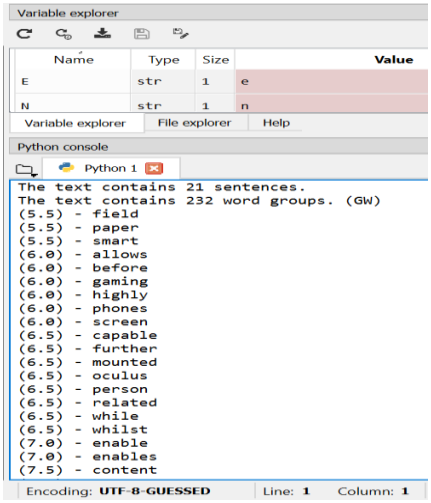

Fig 2 : Length of the Sentences and words.

3.9 Sentence ranking chart.

Based on voting models we will describe sentence ranking mechanism where the sentence in a document are ranked based on score given below as in chart(1)

Sentences (s)	score	Rank
1	2.0	4
2	5.3	1
3	2.7	3
4	4.4	2

Chart 1 : Sentence Ranking Chart

We can retrieve all the sentence closely rated to sentence s by using similarity check. The similar sentences is listed as eligible if and only if using value greater than give threshold value.

Now we will be having own list of tokens. We get a map of how the number of votes depend on the similar sentences that are retired using this list of votes.

3.10 Equations:

We employ and models the first model called as Reciprocal rank, where it computes sum of reciprocal rank of each sentences voting the candidate sentences.

Score of candidate=(sum of all reciprocals of rank).

Another model called as *Combsum*, here we compute sum of score of each sentence voting the candidate sentence.

Score of candidate= sum of all scores(s).

Using the voting model, as mentioned above we obtain the sentence scores. The top ranking sentences extracted and included in the summary.

The machine which accepts the input data will be the form of paragraph format. we basically divide paragraph or document into set of sentence block. The machine performs operation as mentioned above and asks the user to the enter the required amount of summary in terms of percentage. Hence the machine recognizes sentences based on comma separated. It ranks the sentences based algorithms and approaches as discussed above. And gives final summary. A sample the text document summary is shown in Fig 3.

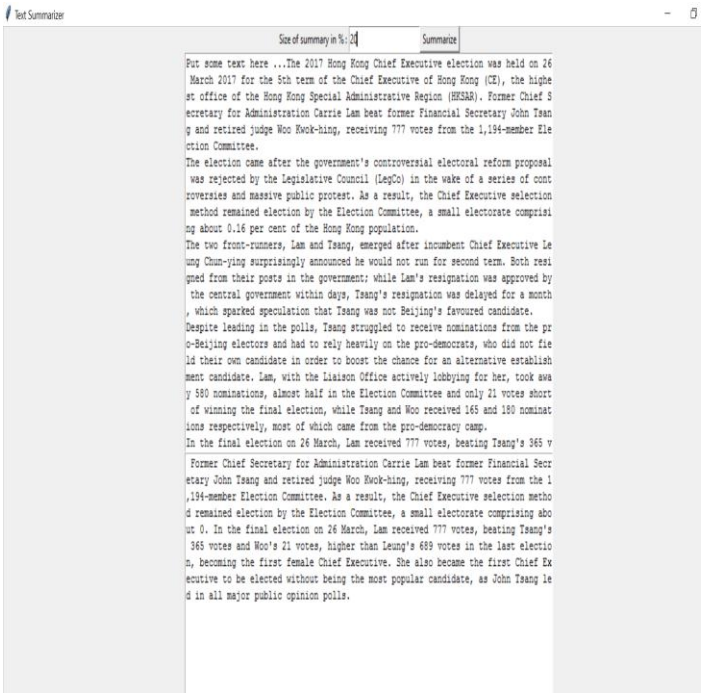

Fig 3 : Summary Output

4. CONCLUSIONS

The quantity of the data which is being generated has expressed for an effective mechanism and detailed summarization schemes for improvising the quality of data. Though this field is underway of improvising the process , a lot of work on optimization has to take place. The paper basically provides an insight of the summarization process by considering the given text data in the paragraph manner and by opting the required percentage of summary, summarized data will be generated as output. Research on this field will continue as it hasn't completed and there will be even more efforts in improving the process.

REFERENCES

[1] Neelima Bhatia and Arunima Jaiswal, "Automatic Text Summarization and it's Methods - A Review "978-1-4673-8203-8/16,2016,IEEE.

[2] "A Noval concept for extraction based text summarization", by Dipti Pawar, S.H.Patil. Vol. 5(3), 2014, 3301-3304.

[3] Yogan Jaya Kumar and Ong sing Goh, "Voting Models for Summary Extraction from Text Document",978-1-2799-6541-0/14,2014,IEEE.

[4] Dharmendra Hingu and Deep Shah , "Automatic Text Summarization of Wikipedia Articles",978-1-4799-5522-0/15,2015 International Conference on Communication, Information & Computing Technology (ICCICT), Jan. 16-17, Mumbai, India.

[5] Harry Zhang, "The optimality of navie bayes", University of New Brunswick, Canada

[6] "Learning to rank from structures in Hierarchical Text classification" by Richard Johansson and Alessandro Moschitti, University of Gothenburg, Sweden.

[7] Zhang Pei-ying , Li Cun-he , "Automatic text summarization based on sentences clustering and extraction", 978-1-4244-4520-2/09/,2009,IEEE.

[8] Mitsuru Ishizuha, "Keyword extraction from a single document using word co-occurrence statistical information".