# Extractive Approach For Query Based Text Summarization

Gayathri Venu Madhuri Chandu[1], Amritha Premkumar[2], Sai Susmitha K[3], Nalini Sampath[4]
[1,2,3,4]Amrita School of Engineering, Bengaluru, India

*Abstract*— **The last few years have seen a tremendous surge in the information that is being dumped online. In this digital world every organization have their respective website which gives a detailed knowledge about them to the public. Considering organizations like educational institutions, their official websites provide all the necessary information from admissions to research works carried out. Prospective students, parents, researchers and academicians refer the website to in order to get relevant answers to their query. In order to retrieve an answer for their query one to has to spend a lot of time searching through the website, reading through several subpages available and consolidating the relevant information. This paper presents a model to retrieve concise and irredundant answers to various questions / queries regarding an educational institution -Amrita School of Engineering. Our model uses various Natural Language Processing (NLP) based techniques for text summarization to give appropriate results. Hybrid similarity measure and clustering algorithm are used for retrieving relevant data and removing redundancy respectively. Our model was tested by many users and the results were accurate in 86% of the cases.**

***Keywords—NLP, Text Summarization, Clustering, Semantic similarity, Word-order similarity, Hybrid Similarity Measure***

## I. INTRODUCTION

In this digital era, the amount of online data is tremendous. According to [1], each day around 380 websites are being created that has information relevant to a particular organisation or an individual. The exponential growth of educational institutions over the recent years has a major contribution to the billions of websites that are present currently. Prospective students and parents go through the university websites to find answers to their queries about the admission process, faculty, research, labs , facilities etc. Due to the number of sub pages and vast information available on these sites, it becomes difficult for one to search required answer to their query. Moreover, consolidating the relevant information from different parts of the website also becomes very tedious while surfing. There might be Frequently Asked Questions (FAQ'S) for many websites but the number of the queries is limited. To solve this problem we have built an efficient query-based text summarisation [2] model adapted to university relevant data. Query-based text summarization is the way to extract relevant information which answers the query referring the original document.

The retrieved summarized answer is short within limited words [3]. It can be classified into two classes: abstractive and extractive. Abstractive summarization first represents the meaning of the entire documents in an abstract form in a semantic space and then synthesizes text using text generation techniques. On the other hand, extractive summarization techniques collect sentences that are most relevant to the question and stitch these sentences together to form a readable summary [4-7]. While abstractive methods attempt to maintain the coherence at the global, it fails to maintain the same at the local level. These particular pros and cons are reversed in the case of extractive techniques since the sentences are not generated but are picked from the manually written text. Our model uses extractive summarization technique.

In this model, we have worked on improving existing methods on extractive summarization in order to generate human readable, coherent, non-redundant and precise query-oriented summaries. The dataset used in our work is prepared by collecting the data from all the available websites of Amrita School of Engineering. Our model has two phases i. Retrieving query relevant sentences from the original text, ii. Removing redundant sentences from the relevant data to generate summarized answer. Various NLP based similarity measures [8] have been used for i. We have used a hybrid similarity measure that captures the semantics and word-order between the query and the answer. Clustering approaches [9] are used for redundancy removal,

ii. We have fine-tuned agglomerative clustering algorithm to adapt to the university relevant dataset to eliminate redundancy in the retrieved answers. Our overall model pipelines these two stages. Section II contains work related to text summarization. Section III explains our implementation. Section IV shows our results. Section V contains the conclusion and related work.

## II. RELATED WORK

Naveen et al. [10] proposed a model for multi document query-based summarization. This model is divided into three phases' i. retrieval, ii. clustering, iii. summarization. They have used cosine similarity for answer retrieval phase. For the clustering phase they presented a novel hybrid clustering technique in extension to the regular hierarchical agglomerative clustering (HAC) and K-means clustering. Highest ranked sentence from each cluster was then retrieved by their model to generate the summary during the last phase. Their model was trained and tested on the Document Understanding Conference (DUC) dataset. They showed that

978-1-7281-1772-0 ©2019 IEEE

their hybrid clustering approach produced more precise and effective results over the regular techniques.

In the work proposed by Shiva Kumar et al. [11], on “Text summarization using clustering technique and SVM technique”, used DUC dataset with a cascaded clustering approach for redundancy removal. Relevant sentences from documents were extracted using cosine similarity of query. An SVM cascaded clustering algorithm was implemented to generate summary from the extracted sentences. The accuracy of the model was measured by comparing the number of tokens in the input and the output summary. They also stated that summarization techniques with clustering are very useful for redundancy removal and sentence simplification.

S.Siji Rani et al. [12] built a graph based model for single document summarization. Their proposed technique obtains summaries using two scoring systems namely undirected graph-based scoring and word frequency-based scoring at paragraph and document levels respectively. They concluded that graph-based model for summarization produce promising results.

A work done by N. Lalithamani et al. [13] on “Discussion Summarization”, proposes a novel clustering that is two tired. They used Yahoo questions with relevant topic information from the website as their dataset for the model. In the first tire of clustering, the sentences are grouped based on topics. In the second tier, clustering is applied within each of these topic clusters at a sentence level. The retrieved sentences are ranked using their proposed Bi Type graph model. The resultant summaries were comparable to the existing summarization outputs.

Vignesh Thiagarajan et al. [14] proposed a summarization model with the use case to reduce manual work for many users of different websites. They made use of cosine similarity to extract query relevant answer sentences. In addition to rank the retrieved sentences they have used a graph-based text rank algorithm to prioritize sentences and generate summary.

All the summarization works discussed above have not focused on improving the similarity phase to retrieve query relevant sentences. We proposed a hierarchical hybrid measure that uses cosine similarity, semantic similarity and word order similarity. We have also fine-tuned agglomerative clustering to generate better irredundant sentences which are concatenate to generate meaningful summary.

## III. Implementation

In this section, we describe in detail the implementation steps of our model. Our data collection mechanism from Amrita School of Engineering online websites along with the necessary preprocessing steps are briefly described initially. The two phases of our model - i. retrieving query relevant sentences from the original text. ii. removing redundant sentences from the relevant data are described in detail later.

### *A. Data Collection*

We have collected all the unstructured data from various online websites of Amrita School of Engineering [15-18]. We have manually converted all the structured data (tables, lists etc.) from the websites into unstructured text format. All the resulting sentences have been concatenated to one sentence per line into a text document. Table I gives the statistics of the data.

Table I. DATA STATISTICS

| Data Statistics | Count |
|---|---|
| Number of sentences | 2334 |
| Maximum length of sentence (words) | 36 |
| Minimum length of sentence (words) | 5 |
| Average length of sentence (words) | 10 |

### *B. Preprocessing*

The input text is first normalized. Normalization of text includes converting all the words into lower case, converting numbers into words or removing numbers, removing punctuations, removing whitespaces, expanding abbreviations etc. Stop words are also removed during this process. Stop words are those words that don't add to the substance of the content however help in the structure of the printed information. Evacuation of stop words chiefly helps in lessening the span of the corpus.

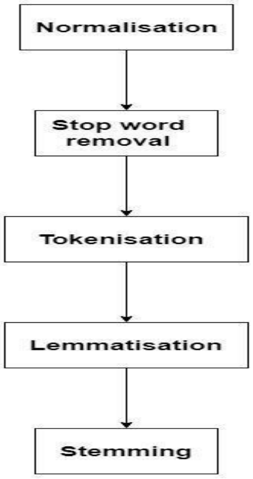


Fig.1 Pre-Processing Steps

Tokenization is the way toward breaking a flood of words or sentences or sections into words, expressions, images or other important components called as tokens. The primary point of tokenization is to investigate the words in a sentence. The token that are delivered after tokenization are utilized for further handling. These tokens are again filtered out using lemmatization and stemming. Lemmatization is utilized to lessen the inflectional structures to a typical base structure. This procedure utilizes lexical learning bases to get the right base type of words. Stemming is a procedure of decreasing words to their pledge stem, base or root structure. Fig 1. shows preprocessing steps.

### C. *Retrieving query relevant sentences from the original text*

To identify the sentences that are relevant to the input query we need to check the similarity between the two. This similarity check can be done in many ways [6]. One of the methods of finding the similarity measure is the cosine similarity. The formula for cosine similarity is given in the fig 2.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Fig. 2 Cosine Similarity Formula

With cosine similarity we convert sentences into vectors. The angle that is suspended between two arrays or vectors is measured by this similarity. One way to do this is to use bag of words with either TF or TF-IDF. This measure is available in the sklearn library of python, which can be used by importing the measure.

Another similarity measure is word-order similarity which measure similarity based on word order of the sentence and query. Sentences containing similar words yet in various requests may result in altogether different implications. It would be simple for people to translate such sentences. Be that as it may, for machines distinguishing the contrast between such pair of sentences isn't so natural. In such cases utilizing this word request comparability measure is valuable. Fig 3. Shows the formula to compute word order similarity.

$$\text{Word Order Similarity (s1, s2)} = (1 - ( \textstyle\sum_{i=0}^{n} |r1[i] - r2[i]| \ / \ \sum_{i=0}^{n} |r1[i] + r2[i]| ))X\ 100$$

Fig. 3 Word Order Similarity Formula

In Fig 3, r1 and r2 are the two vectors where r1 alludes to vector/array consisting positions of the words in the sentence s1 and r2 is a vector/array consisting the positions of the tokens in the sentences s2 with respect to s1.

Semantic measure is the key similarity measure that computes the semantic closeness of the query and sentence. Semantic similarity expects to discover the comparability between two sentences dependent on the importance of the individual words in the sentences. Significant words in the sentences, for example, things, action words, descriptive words, verb modifiers are supplanted by their equivalent words and afterward, a compatibility check is performed. The steps involved in calculation of this similarity are parts of speech tagging, removing stop words, searching for synonyms for every word and calculating the similarity between the sentences and query synonyms. So here for POS labeling, we utilize standard Stanford NLP POS tagger and for distinguishing the synonyms or equivalent words we utilize standard WordNet lexicon.

In our model, we use a hybrid measure that is a combination of these three measures to identify the sentences in the input text that are similar to the query. This hybrid measure is hierarchical with two tiers. In the first tier the cosine similarity is measure between query and input sentence. Our experimentation showed that a threshold of 0.7 at this level yields best outputs. All the sentences passing this threshold are then given as input to the second tier where a weighted sum of semantic and word order similarities is used. Based on our analysis, we set the weights of semantic similarity and word-order similarity as 0.85 and 0.15 respectively. Based on our analysis of academic data in online websites, we found that semantic variance is more frequent than word-order variance. Hence a higher threshold was set accordingly. Fig 4. shows our Hybrid similarity measure.

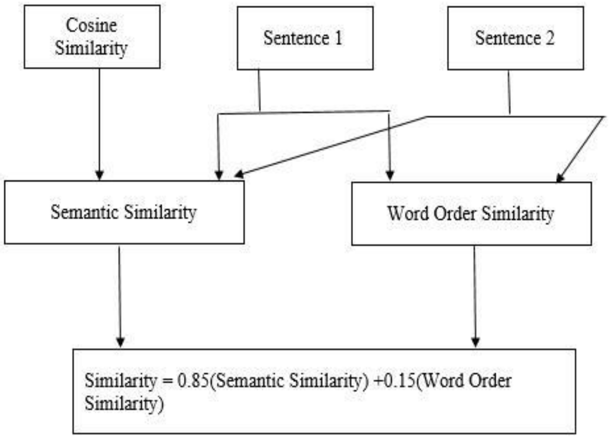


Fig. 4 Similarity measure of the Model

### D. *Removing redundant sentences from the relevant data*

When sentences are picked out after passing the threshold value, there can arise a problem of extracting many one or more sentences of the same meaning. This leads to redundant sentences in our output. To remove this redundancy, we use clustering methods. We have used DBSCAN and Agglomerative clustering algorithms in our model. We have implemented DBSCAN algorithm and analyzed the output to get the correct cluster count that has to be given as an input to Agglomerative clustering. With our experimentation over small part of the data, we have made a conclusion to set the cluster count to 90% of the number of clusters of DBSCAN [19]. As a novel approach, in order to reduce the clustering based on proper nouns we have given lesser weights to the nouns in the sentence while clustering. Fig 5. Shows our clustering module.

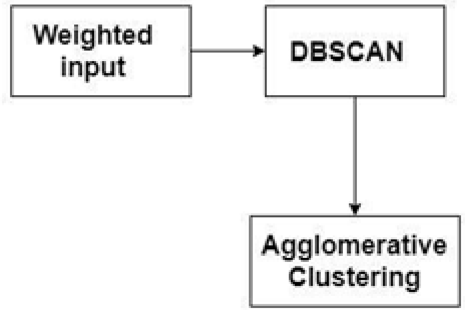


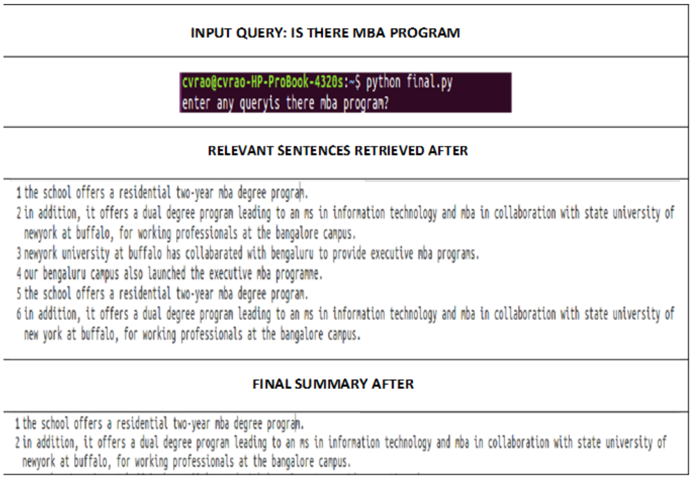
Fig 5. Clustering Module

### E. *Output: Summarized Text*

After all the above four steps are performed we will be able to generate the appropriate summarized text for a given query. Since we are dealing with query based extractive text summarization the output would contain sentences that are present in the corpus. All the relevant sentences after removal of redundancy are printed out as output.

## IV. Results

The model built correctly identifies the relevant sentence to the query and then removes redundancy from the resulting sentences. Fig 6. Shows the output for an example query "*Is there mba program"*. Our model works efficiently in many cases, but there are some cases where the model fails. Fig 7. shows an example where relevant sentences are correctly retrieved but some of these potential sentences are not added to the summary after the redundancy module is implemented.



**INPUT QUERY: IS THERE MBA PROGRAM**

```
cvrao@cvrao-HP-ProBook-4320s:~$ python final.py
enter any queryis there mba program?
```

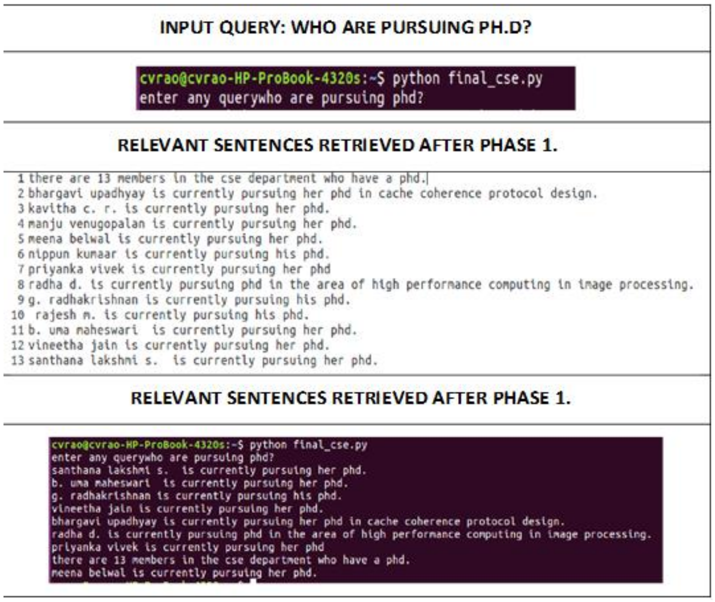
**RELEVANT SENTENCES RETRIEVED AFTER**

```
1 the school offers a residential two-year mba degree program.
2 in addition, it offers a dual degree program leading to an ms in information technology and mba in collaboration with state university of
  newyork at buffalo, for working professionals at the bangalore campus.
3 newyork university at buffalo has collabarated with bengaluru to provide executive mba programs.
4 our bengaluru campus also launched the executive mba programme.
5 the school offers a residential two-year mba degree program.
6 in addition, it offers a dual degree program leading to an ms in information technology and mba in collaboration with state university of
  new york at buffalo, for working professionals at the bangalore campus.
```

**FINAL SUMMARY AFTER**

```
1 the school offers a residential two-year mba degree program.
2 in addition, it offers a dual degree program leading to an ms in information technology and mba in collaboration with state university of
  newyork at buffalo, for working professionals at the bangalore campus.
```

Fig 6. Output for query "Is there mba program?"



**INPUT QUERY: WHO ARE PURSUING PH.D?**

```
cvrao@cvrao-HP-ProBook-4320s:~$ python final_cse.py
enter any querywho are pursuing phd?
```

**RELEVANT SENTENCES RETRIEVED AFTER PHASE 1.**

```
 1 there are 13 members in the cse department who have a phd.
 2 bhargavi upadhyay is currently pursuing her phd in cache coherence protocol design.
 3 kavitha c. r. is currently pursuing her phd.
 4 manju venugopalan is currently pursuing her phd.
 5 meena belwal is currently pursuing her phd.
 6 nippun kumaar is currently pursuing his phd.
 7 priyanka vivek is currently pursuing her phd
 8 radha d. is currently pursuing phd in the area of high performance computing in image processing.
 9 g. radhakrishnan is currently pursuing his phd.
10  rajesh m. is currently pursuing his phd.
11 b. uma maheswari  is currently pursuing her phd.
12 vineetha jain is currently pursuing her phd.
13 santhana lakshmi s.  is currently pursuing her phd.
```

**RELEVANT SENTENCES RETRIEVED AFTER PHASE 1.**

```
cvrao@cvrao-HP-ProBook-4320s:~$ python final_cse.py
enter any querywho are pursuing phd?
santhana lakshmi s.  is currently pursuing her phd.
b. uma maheswari  is currently pursuing her phd.
g. radhakrishnan is currently pursuing his phd.
vineetha jain is currently pursuing her phd.
bhargavi upadhyay is currently pursuing her phd in cache coherence protocol design.
radha d. is currently pursuing phd in the area of high performance computing in image processing.
priyanka vivek is currently pursuing her phd
there are 13 members in the cse department who have a phd.
meena belwal is currently pursuing her phd.
```

Fig 7. Output for query "who are pursuing phd?"

It is very clear from the above output that a few faculty names are not displayed in the final summary though they displayed in relevant sentences. Also, the sentence "There are 13 members in the cse department who have phd." Can be displayed in the beginning of the summary. this an ordering module can be built based on the context of the query. Finally, we tested our model with many users. We have classified the results of our model into 4 categories:

i. Relevant answers without redundancy
ii. Relevant answers with redundancy
iii. Irrelevant answers without redundancy
iv. Irrelevant answers with redundancy

The results of our model are shown in Fig 8. Our model worked accurately in 86% of the cases.

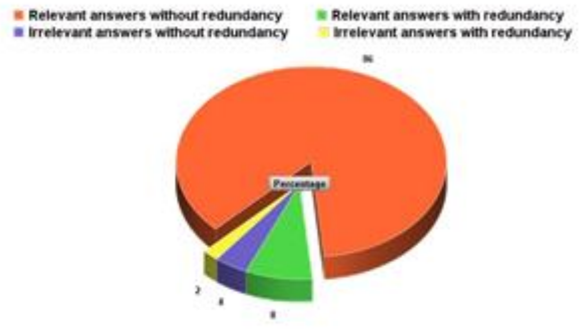


Fig. 8 Results

## V. Conclusion And Future Scope

With the ever increasing number of online websites, there arises a need for new models to reduce the manual work of users in surfing the website for the required data. In this paper we have collected data related to Amrita School of Engineering from various online websites available. Effective query-based text summarization models are needed to satisfy the cause. We have proposed a two-phase model that uses a hierarchical hybrid similarity measure to retrieve relevant sentences and a clustering module to remove redundancy in the extracted sentences. The summaries generated by our model has shown 86% accuracy. As a future scope of the work, this model can be extended to many online websites and blogs. A ranking module can be incorporated to prioritize the sentences in the summary.

## References

[1] https://www.millforbusiness.com/how-many-websites-are-there/

[2] Babar, Samrat & Tech-Cse, M & , Rit. (2013). Text Summarization:An Overview.

[3] Damova, Mariana & Koychev, Ivan & Ontotext, Sofia & , Bulgaria. (2019). Query-Based Summarization: A survey.

[4] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, “Text Summarization Techniques: A Brief Survey,” ArXiv e-prints, 2017

[5] Nadeen M. Abdelaleem, H. M. Abdal Kader, and Rashed Salem ,”A Brief Survey on Text Summarization Techniques”, I.J. of Electronics and Information Engineering, Vol.10, No.2, PP.103-116, June 2019

[6] Allayari, Mehdi and Saeid Safaei. “Summarization Techniques : A Brief Survey.” (2017)

[7] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, Chennai, 2017, pp. 1-6.

[8] Krishna R, Kumar SP, Reddy CS (2013) A hybrid method for query based automatic summarization system. Int J Comput Appl 68:39–43

[9] Nisha and Puneet Jai Kaur, "A survey of clustering techniques and algorithms," *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, 2015, pp. 304-307.

[10] G. K. R. Naveen and Prof. Prema Nedungadi, “ Query-based Multi-Document Summarization by Clustering of Documents ”, in Proceedings of the 2014 International Conference on Interdis ciplinary Advances in Applied Computing, 2014.

[11] K. Mab Shiva Kumar and Soumya, Rab, “Text summarization using clustering technique and SVM technique”, International Journal of Applied Engineering Research, vol. 10, pp. 25511-25519, 2015.

[12] S. Rani S., Sreejith, K., and Sanker, A., “ A hybrid approach for automatic document summarization”, in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017.

[13] N. Lalithamani, K. Alagammai, Kolluru Kamala Sowmya, L. Radhika, Raga Supriya Darisi, S. Shanmugapriya, “ Discussion Summarization”, International Journal of Recent Development in Engineering and Technology, Volume 2, Issue 1, January 2014

[14] N. Lalithamani, “Text Summarization”, Journal of Advanced Research in Dynamical and Control Systems, vol. 10, no. 3, pp. 1368-1372, 2018.

[15] https://www.amrita.edu/school/engineering/about

[16] https://www.amrita.edu/faculty? field_ faculty_department_tid=38&field_faculty_designation_tid=All&field_ faculty_campus_tid=55&fi eld _faculty_department_main_tid=101&field_center_name_tid=All

[17] https://www.amrita.edu/life-amrita-university

[18] https://www.amrita.edu/school/engineering/bengaluru/computer-science

[19] V. K. Viswanath, C. G. V. Madhuri, C. Raviteja, S. Saravanan and M. Venugopalan, "Hadoop and Natural Language Processing Based Analysis on Kisan Call Center (KCC) Data," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, 2018, pp. 1142-1151.