Textual Summarization of Text and Multimedia Data using LDA Algorithm

Prajakta Bharat Deshmukh
Computer Department
MMCOE
Pune, India

S. S. Shiravale
Computer Department
MMCOE
Pune, India

ABSTRACT
To generate a summary lots of efforts have been taken in past years for the events such as Meetings, Sports-clips, Pictorial Storylines, Movies, Social media contents. Natural Language Processing (NLP) is a basic Automatic text summarization application which goals to summarize a given text into a compressed form. Over the year the fast growth in multimedia data across the internet, demands summarization from the asynchronous data that is the combination of image, text, video, and audio. We have describe an multi-modal summarization framework that uses the techniques of OCR, NLP and speech processing examine the information contained in the statics and to enhance the aspect of multimedia summarization.

Keywords
Summarization; Multimedia; Multi-modal; Cross-modal; Natural Language Processing; Computer Vision; OCR Technique; Automatic Speech Recognition.

1. INTRODUCTION
Text summarization plays an essential role in our everyday life and has been studied for several years. From information retrieval to text mining, we are constantly exposed to text summarization. As the use of multimedia data is constantly increasing every past time, it is getting difficult for the user to analyze and obtain efficient knowledge from this huge data. Multimodal summarization (MMS) can give users with textual summaries that can help the user to obtain the significance of multimedia data. This textual summary can be provided in a short period of time, without the need of reading the entire documents or watching videos from start to end. When summarizing multimedia data, it consists of synchronized text, speech, and image. For Pictorial-Storylines summarization, the input is in the form of images with captions. There is no existing application that generates the summary that contains asynchronous information into the documents. Intuitively, readers can grasp the significance of the event more easily by scanning the image or the video than by only reading news or documents.

In this work, the demonstration of an MMS system that can provide users with textual summaries to help to gain the basics of asynchronous multimedia data in a short time without reading documents or watching videos from start to end. The main objective of this work is to combine the NLP (Natural Language Processing), CV (Computer Vision) and ASR (Automatic Speech Recognition) techniques. By combining the techniques we can explore a new framework for mining knowledge contained in data so that we can improve the quality of summarization. Text summarization included two main procedures for summarizing text, Extractive text summarization, and Abstractive text summarization. In Extractive Summarization it involves the collection of phrases and sentences from the source document to generate new summary. Abstractive summarization includes generating completely new phrases and sentences to capture the meaning of the source document. This approach is more challenging but is also ultimately used by humans.

2. RELATED WORK
The work is inspired by the research: Text Summarization, Optical character recognition, Speech to Text Conversion.

Text Summarization
There is a tremendous amount of textual material, and it is only growing every single day. This data is in unstructured form and the best that can be done to navigate it is to use search and skim the results. In order to capture the prominent details, there is need to reduce the text data to a shorter focused summary. So that user can operate it more adequately as well as check whether the larger document contain the information that they are looking for.

Text summarization uses different methods such as LexRank [1], [14] which builds a graph with nodes and edges, where nodes are sentences and edges are the relationship between them. Graph-based methods [6], [7], [12], [21] that generally used for summarization on assumption that if the sentence is similar to many other sentences in the corpus it is likely to be of greater importance. Extractive-Based summarization uses different linguistic features, such as TF-IDF [10] and sentence position [17], [18] to identify the most prominent sentences in a set of documents.

Optical Character Recognition
Optical character recognition (OCR) [3], [8] is a character identification technique that recognizes text region from images, printed text. When a text document or image of a page from a book is scanned it is converted into a bitmap

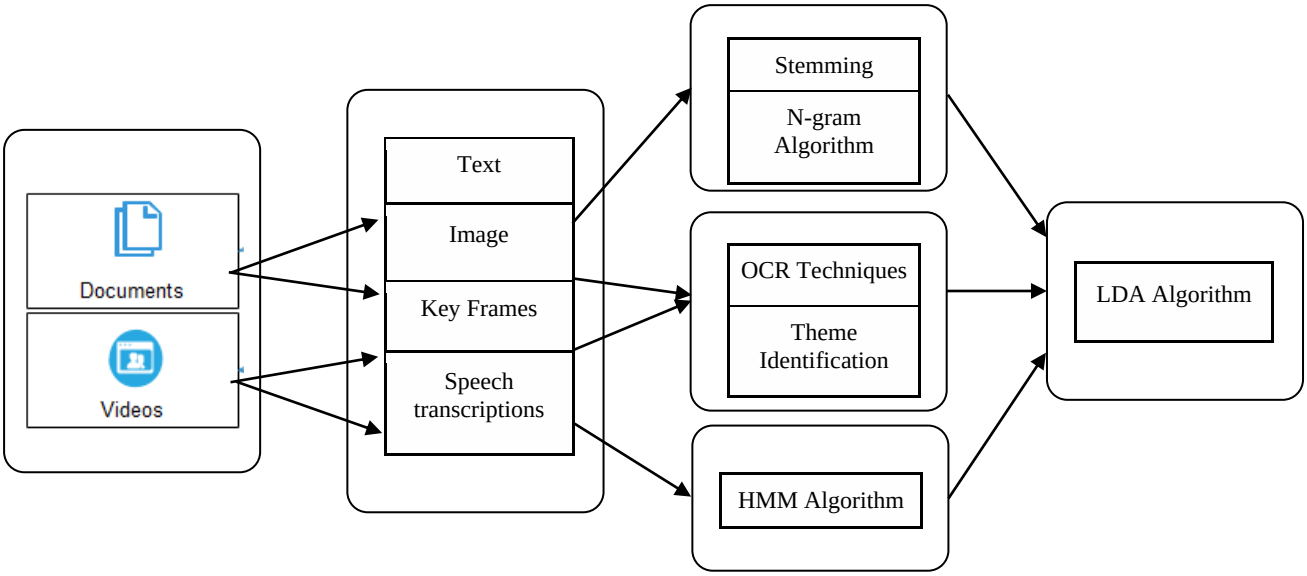

Fig. 1 System Architecture

[2], which is an image of the text. OCR technique matches the light and dark features of the bitmap in order to discover each character. Then these characters are turned into raw text which can be edited easily.

There are three basic steps Segmentation, Feature Extraction, and Classification to achieved character recognition. Segmentation [5], [15] defines components of an image. It is necessary to locate regions in the document that have printed text and are separated from figures and graphics. The most difficult step in pattern recognition is Feature Extraction [16], [19], [20] its an approach to select certain features that identify symbols but leaves the unimportant attributes behind. Next and important step is the Classification [4], [22] that compares the symbol that is extracted from the image to the font, to identify which character it is.

Speech to Text Conversion

Automatic speech recognition (ASR) [13], [23] is also known as Automatic Voice Recognition (AVR), Voice-to-Text or Speech Recognition. ASR is used to identify and process human voice by the use of computer hardware and software-based techniques. It is used to recognize the words a person speaks or to authenticate the identity of the person speaking into the system.

Recognition and translation of spoken language into text. When we speak we create a vibration in the air these are the analog signals. We have to convert these analog signals into digital signals (ADC) [11] that computer can understand. To perform the conversion samples the sound by taking a precise measurement of the wave at frequent intervals. Then perform pre-processing steps and signals are divided into small segments. Match these segments to it's know phonemes [9], [24]. These phonemes then form the word.

3. PROPOSED METHODOLOGY

3.1 Architecture

This paper proposes an approach to generate a textual summary from a set of asynchronous documents, images, audios, and videos, as shown in Fig.1. Because multimedia data are heterogeneous and contain more complex information than that contained in pure text. The MMS Framework shows in Fig. 1. For the audio information contained in videos, obtain speech transcriptions through ASR, and design a method to selectively use these transcriptions. For visual information, including the key frames extracted from videos and the images that appear in documents learn the joint representations of text and images with a neural network; then identify the text that is relevant to the image based on text-image matching or multi-modal topic modeling. In this way, audio and visual information can be integrated into a textual summary by joint optimization. Contribution work is, to design an MMS method that can automatically generate a textual summary from a set of asynchronous documents, images, audios, and videos related to a specific event. Consider four criteria like, salience, non-redundancy, readability, and coverage for visual information that are jointly optimized by the budgeted maximization of sub-modular functions to select the representative sentences. Bridge the semantic gap between the textual and visual data.

3.2 Algorithm

Module 1: Textual Processing

N-gram Algorithm:

N-gram Algorithm:

```
Input: N = Size of N-gram
          Sent = Number of sentences
          ngramList = list to store generated N-grams
Step 1: String [] tokens = sent.split("¥¥S+")
          // sentence split into tokens
Step 2: if (k < (token.length - N + 1) then
Step 3: String S = " " ;
            int start = k ;
            int end = k + N ;
Step 4: for ( int j = start; j < end; j++) do
            S = S + " " + token[j]
```

Step 5: end for

Step 6: ngramList.add(S) ¥¥add N-gram to the list

Step 7: k++

Step 8: end if

Step 9: return (ngramList)

Module 2: Image Processing

OCR Algorithm:

Process 1: Generating the edge of image
Image generateEdgeImage(Image grayImg)

//Create an *X Y* output image edgeImg

//grayImg is the *X Y* result image created in step 1

Step 1: $x \leftarrow 0; y \leftarrow 0; left \leftarrow 0; upper \leftarrow 0;\ rightUpper \leftarrow 0;$

Step 2: **for all** $pixel_{x,y} \in grayImg$ **do**

Step 3: **if** $(0 < x < X - 1)$ ***and*** $(0 < y < Y)$ **then**

Step 4: $left \leftarrow |pixel_{x,y} - pixel_{x-1,y}|$

Step 5: $upper \leftarrow |pixel_{x,y} - pixel_{x,y-1}|$

Step 6: $rightUpper \leftarrow |pixel_{x,y} - pixel_{x+1,y-1}|$

Step 6: $edgeImg_{x,y} \leftarrow \max(left, upper,\ rightUpper)$
Step 7: **else**

Step 8: $edgeImg_{x,y} \leftarrow 0$

Step 9: **end if**

Step 10: **end for**

Step 11: $edgeImg_{x,y} \leftarrow sharpen(edgeImg)$

Step 12: return($edgeImg$)

Process 2: Localizing text candidates
textRegion[ ] detectTextRegions(Image edgeImg)

// edgeImg is created using process 1

// textRegion is a data structure with 4 fields: x_0, y_0, x_1, y_1

// determineYCoordinates uses the process 3

// determineXCoordinates uses the process 4

Step 1: $Integer[\,]H \leftarrow calculateLineHistogram(edgeImg)$

Step 2: textRegion[ ] $TC \leftarrow$ determineYCoordinates(H)

Step 3:$TC \leftarrow$ determineXCoordinate $(edgeImg, TC)$

Step 4: return(TC)

Process 3: Determining the Y-coordinates of text regions
$textRegion[]\ determineYCoordinate$

$(Integer[]\ H)$

// H is the line histogram, see step 3

Step 1: $textRegion\ rect$;

Step 2: $textRegions[]\ TC; y \leftarrow 1, j \leftarrow 0; insideTextArea \leftarrow false$;

Step 3: **for** $el_y \in H$ **do**

Step4:**if**$\Big((el_y > MinEdges)\boldsymbol{or}\big((el_y - el_{y-1}) > MinLineDiff\big)\Big)$**then**

Step 5: **if not** $insideTextArea$**then**

Step 6: $rect.y0 \leftarrow y$

Step 7: $insideTextArea \leftarrow true$

Step 8: **end if**

Step 9: **else if**$insideTextArea$**then**

Step 10: $rect.y1 \leftarrow y - 1$

Step 11: **if** $((rect.y1 - rect.y0) > MinLines)$**then**

Step 12: $TC[j] \leftarrow rect$

Step 13: $j \leftarrow j + 1$

Step 14: **end if**

Step 15: $insideTextArea \leftarrow false$

Step 16: **end if**

Step 17: **end for**

Step 18: **return**(TC)

Process 4: Determining the X-coordinates of text regions
$textRegion[]\ determineXCoordinate$

$\begin{pmatrix} Image\ edgeImg, \\ textRegion[]\ TC \end{pmatrix}$

Step 1: $left \leftarrow maxInt, right \leftarrow -1;$

Step 2: **for**$textCandidate_i \in TC$**do**

Step 3: **for all** $pixel_{x,y} \in textCandidate_i$**do**

Step 4: **if** $(edgeImg_{x,y} \neq 0)$**then**

Step 5: **if** $(left > x)$**then**

Step 6: $left \leftarrow x$

Step 7: **end if**

Step 8: **if** $(right < x)$**then**

Step 9: $right \leftarrow x$

Step 10: **end if**

Step 11: **end if**

Step 12: **end for**

Step 13: $textCandidate_i.x0 \leftarrow left$

Step 14: $textCandidate_i.x1 \leftarrow right$

Step 15: **end for**

Step 16: **return**(TC)

Process 5: Generating the text image
$segmentTextRegions(Image\ edgeImg,$

$textRegion[]\ TC)$

// $edgeImg$ is created with Alg. generating the edge image

// TC is the array returned from Alg. $determineXCoordinate$of text regions

Step 1: Image $reducedImg \leftarrow erase(TC, edgeImg)$

Step 2: Image $binaryImg \leftarrow binarize(reducedImg)$

Step 3: Image $gapImg \leftarrow fillGaps(binaryImg)$

Step 4: $TC \leftarrow refineCoordinates\,(edgeImg,\ gapImg, TC)$

Step 5: Image text$Img \leftarrow extractImage\,(grayImg,\ TC)$

Step 6: text$Img \leftarrow enhanceContrast(textImg)$

Step 7: **return**(textImg)

1. Hidden Markov Model (HMM) algorithm for speech recognition:

A HMM is characterized by 3 matrices viz., A, B and PI.

A - Transition Probability matrix $(N \times N)$

B - Observation symbol Probability Distribution matrix $(N \times M)$

PI - Initial State Distribution matrix $(N \times 1)$

Where, N =Number of states in the HMM

M = Number of Observation symbols

After can apply HMM for speech recognition by using following steps:

1. Recursive procedures like Forward and Backward Procedures exist which can compute P (O|L), probability of observation sequence.

Forward Procedure:

Initialization:

$$\alpha_1(i) = \pi_i b_i o_1, \qquad 1 \le i \le N$$

Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}),$$
$$1 \le t \le T-1, 1 \le j \le N$$

Termination

$$P(O|\lambda) \sum_{i=1}^{N} \alpha_T(i)$$

Backward Procedure:

Initialization:

$$\beta_T(i) = 1, \qquad 1 \le i \le N$$

Induction

$$\beta_T(i) = \sum_{j=1}^{N} a_{ij}\ b_j(o_{t+1}) \beta_{t+1}(j),$$
$$T-1 \le t \le 1, 1 \le i \le N$$

Termination

$$P(O|\lambda) \sum_{i=1}^{N} \alpha_T(i)$$

2.The state occupation probability $t(sj)$ is the probability of occupying state sj at time t given the sequence of observations $O_1, O_2, \ldots, O_N$.

3. Baum-welch algorithm for parameter re-estimation.

Text Summary

Finally after converting the data into the text form, from image and video and completing the preprocessing steps for text. The text is forwarded to the textual summarization algorithm that is the LDA algorithm.

Latent Dirichlet Allocation (LDA) Algorithm:

The first and most important principle, LDA provides a general model that describes how the documents in the dataset were created. In the context, a dataset is a collection of D documents. A document is nothing but the collection of words. So the general model describes how each word in the document is obtained. Initially, let's consider that we know the K topic distributions for the dataset, this implies K multinomials include V elements each, where V is the number of terms in the corpus. βi, represents the multinomial for the ith topic, where the size of βi is V, therefore |βi|=V. From the given assumption the processing steps of LDA are as follows:

Steps: For each document:

1.Randomly pick a distribution over topics (multinomial of length K)

2. for each word in the document:

(i) Pick one of the topics through a probabilistic distribution of topics obtained from 1, suppose topic βj

(ii) Probabilistically draw one among the V words from βj

4. RESULT AND DISCUSSIONS

Experiments are done by a personal computer with a configuration: Intel i3 core @ 1.1Ghz, 2GB memory, Windows, MySQL 5.0 backend database, and JDK 1.7. The application is a web application tool used for designing code in Eclipse and execute on the Tomcat server. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. ROUGE-1 score simply refers to the overlap of unigram (each word) between the system and reference summaries. It is a set of metrics used for evaluating summarization in natural language processing. The metrics compare a generated summary with a human-generated summary (reference summary).

Recall = number_of_overlapping_words \ total_words_in_reference_summary

Precision = number_of_overlapping_words \ total_words_in_system_summary

F1- Score = 2 * Precision * Recall \ (Precision + Recall)

Accuracy = words classified correctly \ all words classified

System Summary (system generated): the cat was found under the bedReference Summary (human-generated) : the cat was under the bed

Recall = 6 \ 6 = 1

Precision = 6 \7 = 0.86

F1 - Score = (2 * 0.86 * 1) \ (0.86 + 1)

OCR: Training dataset 450 pages for samples for tesseract training dataset which contains some good quality and poor

quality pages. The testing dataset is the any image file which contains textual part.

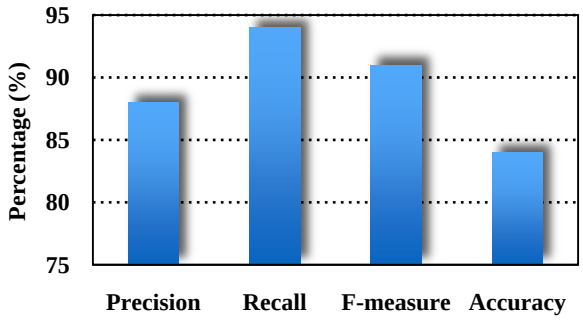

Fig. 2 OCR performance analysis using tesseract

Audio: To build a language model using the CMU Sphinx 4 training tools: the transcription file, the phonetic dictionary, the list of phonemes, and the recordings of English recitations. The "cmudict-en-us.dict" dictionary must update the configure file with the path of the dictionary and the path of the fillser file also in the dictionary component. To load the model must be changed to the path of the building one, as the proposed system covers a large vocabulary so the n-gram model is used here. By the use of the Sphinx ASR system, an in-domain language model was trained using the corpus of phrases. To evaluate the speech recognition systems practically, the word error rate and response time were measured. The WER is working at the word level instead of the phoneme level. Word error rate can then be computed as:

$$WER = \frac{S + D + I}{N}$$

Where,

S is the number of substitutions,

D is the number of the deletions,

I is the number of the insertions,

N is the number of words in the reference.

We are applying the segment-based speech recognition method.

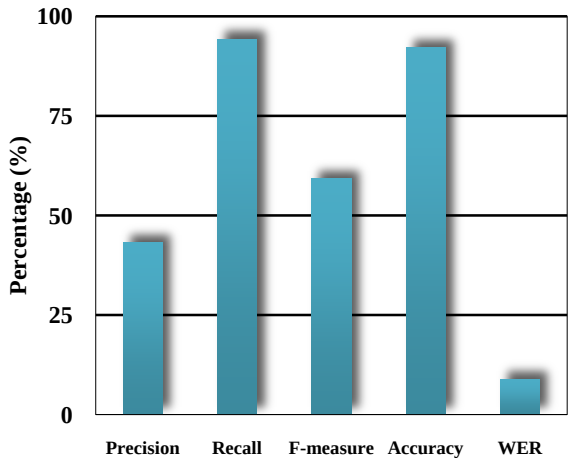

Fig. 3 ASR performance analysis using sphinx

For English Multi-modal Summarization Framework, Fig. 4 shows that when summarizing textual data like, .txt, .doc, .docx etc. file format it gives better accuracy. The image file applies the OCR algorithm for textual part extraction and generates summary. The HMM model of speech transcriptions performs better on audio files. The video file is same work on OCR and ASR system and finally results are shown in fig. 4 using Text Summary Generation Algorithm.

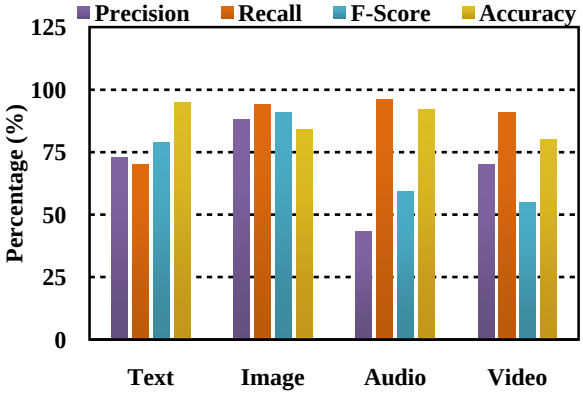

Fig. 4 Performance graph for MMS framework

5. IMPLEMENTATION OUTPUT

The LDA algorithm is applied to the text data to generate the summary. When user upload the file in our system the file goes through different modules. For images present in file, the file is goes through OCR technique where text are detected and retrieve from the image and the retrieve text data is then goes to the textual summarization generation algorithm i.e. LDA algorithm. As for speech to text HMM algorithm is applied and then retrieved text is passed to generate the summary.

Fig. 5 shows the summary generated from the text data where the file containing text was given as input to the system. LDA algorithm was applied to the given text to generate the specified length of summary from the data.

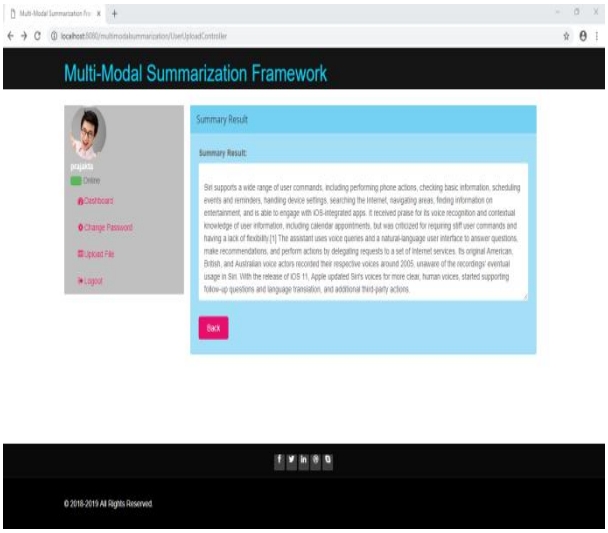

Fig. 5 Summary Result Page

Fig. 6 shows the page of the summary generation. Where the summary is generated from the image given as input to the system. The above output shows the two different sections in which the first section shows the OCR result that is the text extracted from the image and the second section shows the summary generated from the OCR result.

Below Fig. 7 shows the result of speech to text and the associated summary generated from the speech recognition. In this process text is fetched from the audio signals using the HMM algorithm, then on the extracted text LDA algorithm is applied for a textual summary generation.

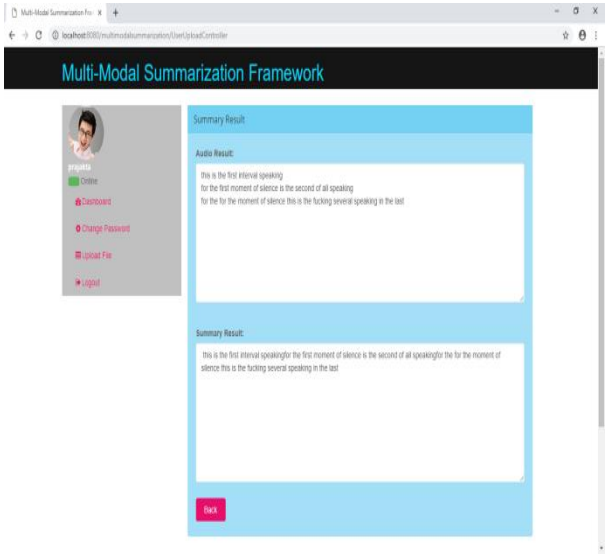

Fig. 6 Audio Result Page

6. CONCLUSIONS

Thus asynchronous MMS model has been implemented successfully, which generates a fixed-length textual summary to represent the fundamental content of the data. The system accepts input of any file extension like, .txt, .docx, .jpg, .mp3, .mp4 etc. A textual summary is generated from the text data given as input to the system. When the input data is in the form of images, then OCR techniques are applied to extract the text from the image. To further generate the summary from this text extracted LDA algorithm is applied. For speech data, the HMM algorithm is applied to fetch text from the audio signals. Thus we have implemented this system to help users easily understand and analyze a large amount of multimodal data.

In the future, the project platform could be extended from application-based to separate websites. Our current system is based on Extractive Summarization in the future we can implement a system based on Abstractive Summarization.

7. REFERENCES

[1] Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, Chengqing Zong, “Read, Watch, Listen and Summarize: Multi-modalSummarization for AsynchronousText, Image, Audio and Video" in IEEE Transactions on Knowledge and Data Engineering, 2019.

[2] Jorge Poco, Angela Mayhua, and Jeffrey Heer, “Extracting and Retargeting Color Mappings from Bitmap Images of Visualizations”, in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, 2018.

[3] A. Chaudhuri et al., “Optical Character Recognition Systems for Different Languages with Soft Computing, Studies in Fuzziness and Soft Computing 352”, Springer International Publishing AG 2017.

[4] Bharath, V., & Rani, N. S., “A font style classification system for English OCR”, International Conference on Intelligent Computing and Control (I2C2), 2017.

[5] Mohammad Azim Ul Ekram, Anjani Chaudhary, Ashutosh Yadav, Jagadish Khanal, Semih Aslan, “Book Organization Checking Algorithm using Image Segmentation and OCR”, in IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 2017.

[6] H. Li, J. Zhang, Y. Zhou, and C. Zong, “Guide-rank: A guided ranking graph model for multilingual multi-document summarization,” in International Conference on Computer Processing of Oriental Languages. Springer, pp. 608–620, 2016.

[7] X. Zhou, X. Wan, and J. Xiao, “Cminer: Opinion extraction and summarization for chinese microblogs,” IEEE Transactions on Knowledge & Data Engineering, vol. 28, no. 7, pp. 1650–1663, 2016.

[8] Noman Islam, Zeeshan Islam, Nazia Noor, “A Survey on Optical Character Recognition System”, in Journal of Information & Communication Technology-JICT vol. 10 no. 2, 2016.

[9] Zhong-Qiu Wang, Yan Zhao, De Liang Wang, “Phoneme-specific speech separation”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[10] Hans Christian, Mikhael Pramodana Agus, Derwin Suhartono, “Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)”, in ComTech Vol. 7 No. 4, 2016.

[11] Yang, Z., Jin, J., & Wang, M., “A signal processing method using pulse-based intermediate values for delta-sigma analog-to-digital conversion.”, IEEE International Conference on Digital Signal Processing (DSP), 2015.

[12] P. Goyal, L. Behera, and T. M. Mcginnity, “A context based word indexing model for document summarization,” IEEE Transactions on Knowledge & Data Engineering, vol. 25, no. 8, pp. 1693–1705, 2013.

[13] Preeti Saini, Parneet Kaur, ”Automatic Speech Recognition: A Review”, in International Journal of Engineering Trends and Technology Vol. 4, no.2, 2013.

[14] G. Erkan and D. R. Radev, “LexRank: Graph-based lexical centrality as salience in text summarization,” Journal of Qiqihar Junior Teachers College, vol. 22, p. 2004, 2011.

[15] Koppula, V. K., & Negi, A., “Fringe Map Based Text Line Segmentation of Printed Telugu Document Images.”, in International Conference on Document Analysis and Recognition, 2011.

[16] Bilal Bataineh, S. N. H. S. Abadullah, Khairudin Omar., "A Statistical Global Feature Extraction Method for Optical Font Recognition, "presented at the LNAI the 3rd Asian Conference on Intelligence Information and Database Systems (ACIIDS 2011), 2011.

[17] Y. Ouyang, W. Li, Q. Lu, and R. Zhang, “A study on position information in document summarization,” in COLING, pp. 919–927, 2010.

[18] V. Varma, V. Varma, and V. Varma, “Sentence position revisited: a robust light-weight update summarization ’baseline’ algorithm,” in International Workshop on Cross Lingual Information Access: Addressing the

Information Need of Multilingual Societies, pp. 46–52, 2009.

[19] S. N. H. S. Abdullah, et al., "license plate recognition based on geometry features topological analysis and support vector machine, " 2007.

[20] U. Bhattacharya, B. K. Gupta and S. K. Parui, "Direction Code Based Features for Recognition of Online Handwritten Characters of Bangla", in Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[21] X. Wan and J. Yang, "Improved affinity graph based multi-document summarization," in NAACL, pp. 181–184, 2006.

[22] Gupta, G., Niranjan, S., Shrivastava, A., & Sinha, R., "Document Layout Analysis and Classification and Its Application in OCR.", in 10th IEEE International Enterprise Distributed Object Computing Conference Workshops, 2006.

[23] Shamma, S., "Relevance of auditory cortical representations to speech processing and recognition." IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.