I J C T A, 9(2) 2016, pp. 547-555
© International Science Press

Block Diagram Components' Information Extraction and Grading of Research Paper based on Graphical Images

R. Siva[1], G. S. Mahalakshmi[2] and S. Sendhilkumar[3]

ABSTRACT

Block Diagrams claim large importance in the conceptual understanding of data. The Text Extraction from block diagrams, the optimized Image processing techniques and the selective extraction of sentences which contribute to the meaning and functionality of the component are key elements involved in the successful execution. The project aims at presenting essential information of the block diagram components in a comprehensive manner, using the Web-Crawling techniques and to grade a research paper based on the textual explanation of graphical images. The images which are the object of interest for grading are graphical images like bar graphs and pie charts. The primary aim is to detect images like bar graphs, pie charts. Next we extract text explaining the graphical image from the research paper. After extracting text, we analyse it to see how good it is based on various metrics and hence grade the paper.

Keyword: Research Paper Grading, Web Crawling, Definition Extraction, VLDB.

I. INTRODUCTION

A Research paper is written after a strenuous research and it involves lot of work than just conveying the idea to convince the audience who need the results. Thus, research paper grading is very important to judge the standard of the research paper. A good research paper should follow the golden rules for research paper and should convey all the required details.

The paper should contain a detailed summary, a crisp abstract, citations, work cited, a good writing style and a logical, organised paperwork. Research paper grading Rubric, an existing and widely used method grades the research papers based on: Organisation, Writing style, key points, Citations, Analysis, Insight and Grammar or Language use. This project aims at grading a research paper based on the explanation given for the graphical images present in it. The dataset is collected from VLDB. The Analysis of Block Diagrams involves each constituent component to be visualized along with its functionality to understand the broad picture. Research Papers range from having high resolution to moderate and sometimes, even low resolution diagrams which affect the image processing stages and hence, result in inaccurate and improper text extraction from the images present. The availability of commercial and open source software which facilitate the image processing procedures provide a good foundation for implementing the necessary subtasks.

Optical character recognition (OCR) is the mechanical or electronic conversion of images of typewritten or printed text into machine-encoded text. It is a common method of digitizing printed texts so that it can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes

[1] Department of Computer Science and Engineering, KCG College of Technology, Karappakam, Chennai 600097

[2] Department of Computer Science & Engineering,College of Engineering Guindy, Anna University, Chennai 600025

[3] Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai 600025

Email: bala465ji@gmail.com[1], gsmaha@annauniv.edu[2], thamaraikumar@annauniv.edu[3]

such as machine translation, text-to-speech, key data and text mining. Using the text processing algorithms and the sentence tokenizing methods, the required text is extracted and used to focus on further analysis in terms of functionality and definition.

The graphical images in the research paper are extracted and then subjected to histogram conversion. Based on continuity of pixels we distinguish graphical images from non - graphical images. The figure number of graphical images is quoted as keywords. The concerned research papers are converted to text files and the required textual explanation of the graphical images is extracted based on keywords. The extracted text is subjected to analysis and based on several metrics we grade the quality of textual explanation for a graphical image.

II. RELATED WORK

(A) OCR Related Framework

Tesseract is an open-source OCR engine that was developed at HP between 1984 and 1994. Tesseract therefore assumes that its input is a binary image with optional polygonal text regions defined. Tesseract was probably the first OCR engine able to handle white-on-black text so trivially [4]. The key parts of the Line-Finding Algorithm are blob filtering and line construction.

Baseline Fitting Algorithm works on the text lines that have been found, and hence, the baselines are fitted more precisely using a quadratic spline. This was another first for an OCR system, and enabled Tesseract to handle pages with curved baselines. Tesseract tests the text lines to determine whether they are fixed pitch. Where it finds fixed pitch text, Tesseract chops the words into characters using the pitch, and disables the chopper and associator on these words for the word recognition step.

(B) Document Summarization

There is scope to improve the NLP preprocessing. Integrating a named entity (NE) parser in pre-processing stage for handling unknown names improve the quality of the mined patterns. The Acquisition of Domain-Specific Patterns [1] revolves around Data Collection, NLP and advanced text processing techniques, Pattern Mining and Scenario –Based Selection.

Summarization Frameworks rely on Conditional Random Fields and Contextual Modelling by Sentence-level, Non-Textual and Textual Features [3]. An automatic “general purpose” text summarization tool would be of immense utility in this age of information overload. Using the techniques used (by most automatic hypertext link generation algorithms) for inter-document link generation, intra-document links are generated between passages of a document. Based on the intra-document linkage pattern of a text, the structure of the text is determined.

(C) Text Simplification

Text Simplification is enabled by Proper Corpus Analysis and Dynamic approach to Text Alignment. The task of text simplification aims to reduce the complexity of text while maintaining the content. Text compressions or simplification should be grammatical, and they should retain the most important pieces of information. The dataset required for testing text Simplification techniques need to be of high quality. The Proceedings of the VLDB Endowment, Vol 7 are utilized for the vast use of diagrammatic representations [6].

(D) Web Crawling

A program which browses the World Wide Web in a methodical, automated manner is called a web crawler. This process is called Web crawling or spidering. Many legitimate sites, in particular search engines, use spidering as a means of providing up-to-date data. Many applications today, are dependent on the internet for dynamic and latest reliable information. For extended meaning and information retrieval, the web is utilized, without which, in-depth analysis of data is impossible. There are many commercial and open-source web-crawling techniques and platforms which ease the process of web-based searches like Scrapy and the Wikipedia Python API.

(E) Text Extraction And Analysis

Gupta *et al.* [9] suggested Text Summarization Extractive Techniques. An extractive summarization to method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences. An Abstractive summarization method consists of understanding the original text and re-telling it in fewer words. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document.

Traditional scientific papers are unstructured documents, which are difficult to meet the requirements of structured retrieval. Hence, how to extract and analyze the structured information of the papers becomes a challenging problem. Jianguo *et al.* [9] proposed a structured information extraction algorithm for unstructured and/or semi-structured machine readable documents.

Another approach given by Chuang *et al.* [7] is where text is extracted based on sentence segmentation. The basic idea here is to separate out units (segmentation by cue phrases). Sentences are broken into segments by special cue markers. Each segment is represented by a set of predefined features. This cue phrase is used here because which it connects two segments and extracts text. The complexity of this approach is O(n).

III. OUR IDEA

The contributions of our paper are: (1) Conversion of images to histogram and detecting graphical images based on continuity of pixels. (2) Analysis of Extracted text based on several metrics and grading the papers based on these metrics. (3) Effortlessly acquire and manage information about the block diagrams in research documents. This project revolves around the modules encompassing Image Processing and Text Processing.

Since OCR can handle only monochrome images, we pre-process the images. The High speed Tesseract OCR engine uses optimized algorithms to perform excellent recognition in short time. Speed can be further improved through multi-threading and optional GPU acceleration. Python Tesseract Leptonica Software is the key to the text extraction from the block diagrams after the image is extracted from the portable document format file.

For more accurate results, the Auto-Spellcheck libraries of Python, namely PyEnchant is used. The Spelling Rectification module is essential due to the discrepancies that occur in the extractions of text from images. Compilation of descriptive text phrases of the components involved in the block diagram uses Python's Natural Language Toolkit (NLTK) for processing the natural language which is available in raw text format.

The input research document is converted to a text file to proceed with processing. Using the NLTK Sentence Tokenizer, the insignificant sentences are eliminated. The grouping of data regarding each component is done by means of the Phrase-Matching Algorithm. Local Search involves the processing of research paper text for the descriptions that occur explicitly. For words that do not have necessary information within the tokenized text, Web-Crawling broadens the horizons from which the meanings and information can be obtained. Web-Crawling is very essential in the extraction of dynamic and more refined information from the Internet and aids the description extraction process for uncommon terms present in the input set of documents presented. Finally, the processed text as a result of local and global search is presented in a consolidated manner which contains the components and their corresponding description for clarity in understanding the importance of each block diagram entity.

It also grades the quality of research paper based on the description that is given for graphical images present in it. The grade is given based on the justification given for explanation of the graphical images. Since graphical images explain difficult contexts with ease, a good explanation for such figures would improve the quality of the research paper. Analysis of such a text which explains the images would give great scope for criticizing the amount of work the researcher has put into the research paper. Thus based on text analysis metrics, we aim to grade the research paper with justification. The dataset is collected from VLDB.

IV. BLOCK DIAGRAM COMPONENTS' INFORMATION EXTRACTION

(A) Text Extraction from Image

Using the Ghostscript software based on an interpreter for Adobe's PostScript and Portable Document Format page description languages, the PDF Image is extracted and stored separately as a png file. Extraction of text involves text region detection, text localization, tracking, character extraction, enhancement, and recognition of the text from a given image. Variations in text may occur because of differences in size, style, orientation, alignment of text. Low image contrast and composite backgrounds serve as a problem during the extraction of text. Using the Python Tesseract OCR Engine, we aim at processing the image with dilation, monochrome filter and improving the image quality. Accurate Text Recognition is required and with enhanced image processing and text detection algorithms, Tesseract OCR can easily recognize difficult documents of poor image quality. Parameters can be used to hint favouring accuracy over speed and training the engine improves the results. The Extraction of the text from the block diagram is shown in Fig.3 which is done using Tesseract and enhanced with the Ghost Script Software. Figure 3 shows the topology of Block diagram components' extraction.

(B) Spelling Rectification

The text that is extracted from the image is processed and prior to Natural Language Processing, the spelling of each component are checked and corrected for appropriate results. Using the Python API for Spell Checking, namely the PyEnchant, the suggestions help in rectifying the errors in the text extracted. The Spelling Rectification Process is improved by adding a domain- based dictionary to increase the efficiency and accuracy of correcting the components extracted, as the additional dictionary expands the collection of possible words that can occur in technical documents and research papers.

(C) Sentence Tokenization

The Research Papers contain text pertaining to the components involved in the block diagram that are required for analysis. The document is searched and the sentences that are related to the components are tokenized for obtaining the object, relation and the subject. Using the Tokenizer package supplied by the NLTK Library, namely the Punkt Sentence Tokenization model of NLTK tokenizer tokenizes the paragraphs into sentences.

(D) Web- Based Refining

For the components without any description in the paper, web-crawling is used for obtaining an appropriate description to widen the scope of search. We restrict the search within the top few links and try to retrieve a suitable meaning of the component. With the help of the keywords obtained in the document, the most appropriate context- based description is coined to complete the consolidation of all the components with their corresponding description. The Wiki API enhances the definitions that are obtained from the known web encyclopaedia, Wikipedia.org.

(E) Sentence Consolidation

Using the phrase-matching algorithm, the essential sentences required for the formulation of the component definition are extracted and presented in a concise form for validation. The local and global searches produce results which are consolidated for the final presentation of the technical terms with their information.

V. GRADING OF RESEARCH PAPERS

The block diagram of the proposed system is given in Figure 1. Each block is explained with the methodology used.

(A) Pre-Processing

Research paper is given as input in the form of PDF. The PDF contains graphical images and no other images like logos, tables etc. The preprocessing step to be done is Image Extraction from the PDF files. These images are extracted using A PDF Image Extractor tool and then the extracted images are stored in a separate folder. This will evaluate to 100% if there are no logos or tables in the PDF format of the research paper.

(B) Histogram Conversion and Image Detection

Inputs given are the images which are extracted in the preprocessing step. An image is converted to histogram using *matlab* functionality. Here logos and tables are converted to histogram as the functionality considers them as images. This will reduce the accuracy. The histogram files are stored in a separate folder. A histogram is non-continuous for graphical image and it is continuous for the rest of the images.

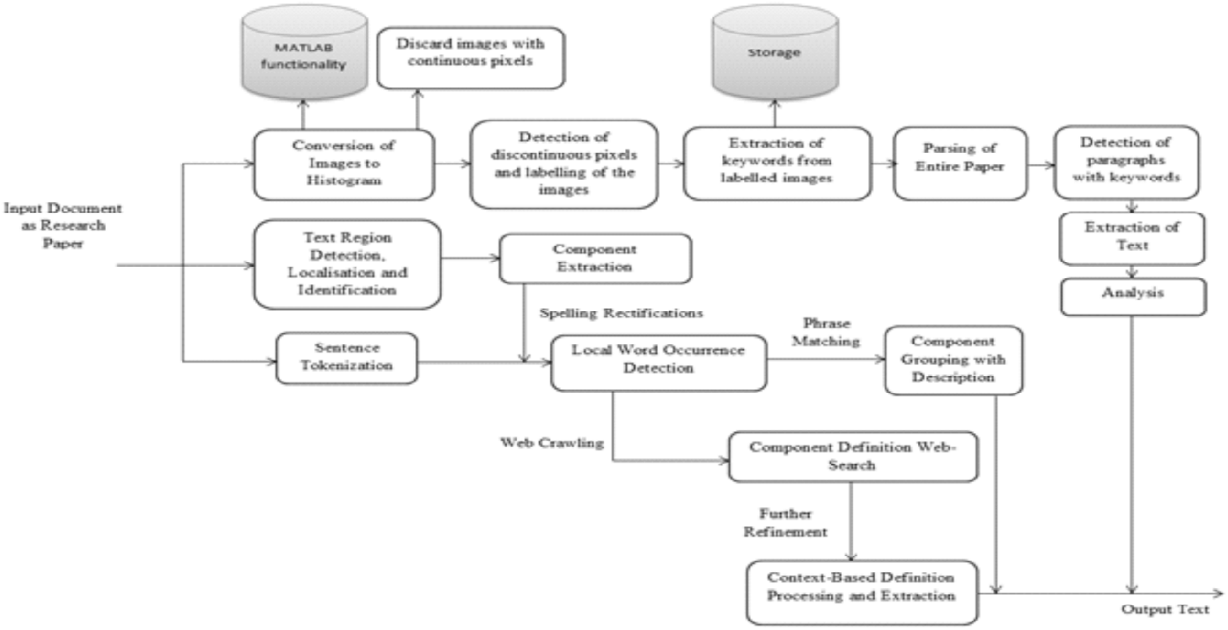

MATLAB functionality
Discard images with continuous pixels
Storage
Input Document as Research Paper
Conversion of Images to Histogram
Detection of discontinuous pixels and labelling of the images
Extraction of keywords from labelled images
Parsing of Entire Paper
Detection of paragraphs with keywords
Extraction of Text
Analysis
Text Region Detection, Localisation and Identification
Component Extraction
Spelling Rectifications
Sentence Tokenization
Local Word Occurrence Detection
Phrase Matching
Component Grouping with Description
Web Crawling
Component Definition Web-Search
Further Refinement
Context-Based Definition Processing and Extraction
Output Text

Figure 1: The diagrammatic representation of functional blocks of grading based on graphical images explanation and the Block diagram component information extraction topology

Based on this, we detect the graphical images. Both normal images and pie-charts have structural similarity due to continuous pixels. So it will assume even a pie chart as a normal image which reduces accuracy. The rest of the graphical images are detected and their figure number is stored.

(C) Conversion of PDF to Text

Research paper is given in the form PDF. It is assumed to be free from captions. PDF file is converted to text files using *pdf* to text conversion java code. The *itext* jar file of java library facilitates the conversion. Output is the converted Text file .The converted text files do not contain the captions given to images.

(D) Text Extraction

The research paper is subjected to a search based on keywords (the word figure), then the related text will be extracted using java code. Input is the Text file along with keywords. Lines of text containing the keywords are obtained as output. This will leave out the explanations which doesn't contain the keywords explicitly .For example in some exceptional cases, the textual explanation will not contain keywords (eg. as shown above), and explanation of the figure would be followed. The system might fail to extract these texts (since no keywords like fig/figure were found). This might increase errors. Even a single-line textual explanation will be extracted, which reduces the efficiency of the system.

(E) Text Analysis and Grading

Several metrics like key-word count, stopword count, line count measure, spelling check, and reference word check and noun format check measure have been implemented to grade the research paper's extracted textual explanation. Graphical images are pictorial explanation. They replace large textual explanation. Hence the line count measure should produce a low value as result.

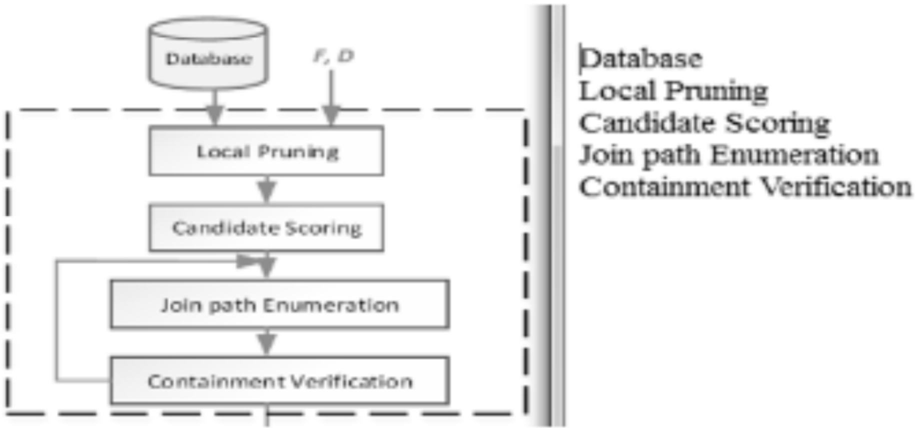
The number of lines should not be too less or too more. If the number of lines explaining the figure is bulky, it reduces the quality. So we use this as one metric for grading. Usually the explanation containing the keywords can refer to the keyword once. Repeated reference annoys the reader. Hence the measure of keyword count should be as low as possible. The next metric used is spell check. We check if the extracted text is free from spelling mistakes. If there are more errors in spelling, the grading decreases eventually.

Graphical images are best explained when there is sufficient information provided about the peaks, slopes, axes, coordinates, min and max values. Such reference words are indexed and checked. If the extracted text contains such words then the explanation is of good quality. The stop word frequency is expected to be lower. The extracted text is subjected to stopword count. Textual explanation of the graphical images is themselves not more than a few lines. Out of these lines, if the stop words count is more, then not much explanation is provided. The stop words are detected and a count is maintained. If the count exceeds threshold the grading decreases as quality faces a fall. More the number of stop words, Lesser the grade is.

VI. EXPERIMENTAL RESULTS

The dataset is collected from VLDB. The Results are calculated based on the block diagram component extraction accuracy and the average number of words in each component present in the dataset, are three on an average. This is the determining factor for the proper sentence extraction and consolidation. To evaluate the text extraction process from the image, it is required to be in high resolution. To do so, the processing of the image must be done with care as it influences the accuracy of the text extraction. Spelling Rectifications require the complete dictionary containing technical terms because the domains of the research papers are scientific and uncommon.

Text Extraction Accuracy (%) = (Number of correctly identified components/Total number of components) X 100 (1)



Database
Local Pruning
Candidate Scoring
Join path Enumeration
Containment Verification

Figure 2: Sample image and text Extraction using Tesseract

To evaluate the text extraction process, the image is required to be in high resolution. The occurrence of certain symbols like arrows, colons, bounding boxes around the components hinders the image processing phase. Pre-processing the image includes the usage of Ghost script and changing the properties of the image. The text-extraction accuracy is improved with the help of Python's Spell-checking library called PyEnchant. The related contents with respect to each component is extracted and presented. This poses the challenge of identifying only the significant points and eliminating the lines which do not define the component. Web Crawling from Wikipedia is done with ease but there are difficulties faced while trying to crawl from google directly. The approach to enhance the significant information extraction is done by training the dataset with necessary keywords. The presence of definitions in sentences, involving the components in the research papers aids in faster consolidation of components and web-crawling determines the time complexity of the presentation of components that do not have local definitions.

The evaluation metrics used for grading part are: Structural similarity measure, pattern matching defect, Accuracy, error rate, precision, recall, and efficiency. A table with all the evaluation metrics and the percentage accuracy is shown in Fig.2. The preprocessing step involves Extraction of all images from the research paper, which is given as input in the form of a PDF. Image Extraction is done using A PDF Image Extractor tool. This will extract all images present in the paper. It considers even tables and logos as images and will extract them. This reduces the precision.

A recall of 76.92% shows that most of the extracted text and images are relevant. Lower value of error rate and pattern matching defect shows that the accuracy of the project is high. Even a single line textual explanation is extracted which makes the efficiency of the system to be 75%. Structural similarity measure is 60% since even pie-charts are considered as normal images and might be ignored during image detection.

VII. CONCLUSIONS AND FUTURE WORK

The conclusion of our experimental analysis suggests that 51% accuracy is obtained with the dataset, being the Proceedings of the VLDB Endowment, Vol 7, 2013-2014. Tesseract [5] is now behind the leading commercial engines in terms of its accuracy. Its key strength is probably its unusual choice of features.

Its key weakness is its use of a polygonal approximation as input to the classifier instead of the raw outlines. With internationalization done, accuracy could probably be improved significantly with the judicious addition of a Hidden-Markov-Model-based character n-gram model, and possibly an improved chopper. Also we have successfully detected graphical images from the dataset and extracted their textual explanation.

Table I
Block Diagram Components Information Extraction Accuracy

No. of Components	*No. of Papers*	*Paper Count*	*No. of components Extracted from block diagram*	*Local Definition Present*	*No. of Components to be Web Crawled*	*Successfully WebCrawled Components*	*Accuracy (%)*
3	1	1	2	2	0	0	66.67
4	3	1	2	2	0	0	50
		2	4	3	1	0	75
5	3	1	3	2	1	1	60
		1	4	3	1	1	80
		1	5	3	2	2	100
6	4	1	2	2	0	0	33.33
		1	3	3	0	0	50
		2	5	4	1	1	83.33
7	4	2	5	4	1	1	71.43
		1	6	3	3	1	57.14
		1	7	6	1	1	100
8	9	3	3	3	0	0	37.5
		3	5	4	1	1	62.5
		1	6	3	3	1	50
		2	8	4	4	3	87.5
9	6	1	4	3	1	1	44.44
		2	5	3	2	2	55.56
		1	6	4	2	1	55.56
		1	7	4	3	1	55.56
		1	9	6	3	2	88.89
10	4	1	7	5	2	2	70
		2	9	6	3	2	80
		1	10	4	6	4	80
11	3	1	5	4	1	0	36.37
		2	11	8	3	2	90.91
12	5	2	6	4	2	1	41.67
		3	9	9	0	0	75
13	4	1	6	6	0	0	46.15
		1	7	6	1	1	53.85
		2	10	9	1	1	76.92
15	3	1	6	5	1	1	40
		2	10	6	4	2	53.33
18	1	1	12	9	3	2	61.11
25	1	1	13	10	3	2	48
TOTAL							
51							**51.54%**

Table II
Evaluation metrics with accuracy in percentage for grading of papers

Evaluation Metrics	*Percentage (%)*
Structural Similarity Measure	60
Pattern Matching Defect	30
Accuracy	80
Precision	83.33
Recall	76.92
Error Rate	20
Efficiency	75

The extraction takes place post the conversion of the research paper from pdf to text format. The analysis of the extracted text is done based on the above suggested metrics and a grading is assigned to the research paper as output. Future work could include better search optimization strategies for combining points for the components with the use of web-crawler so that technical terms can be extracted from authentic sources. To increase the success rate of the text extraction, the image must be processed further before passing it to the OCR Engine.

REFERENCE

[1] Du, M. and Yangarber, R., (2015), April. Acquisition of Domain-specific Patterns for Single Document Summarization and Information Extraction. In*The Second International Conference on Artificial Intelligence and Pattern Recognition (AIPR2015)* (p. 30).

[2] De Nart, D. and Tasso, C., (2013), December. A Keyphrase Generation Technique Based upon Keyphrase Extraction and Reasoning on Loosely Structured Ontologies. In *DART@ AI* IA* (pp. 49-60).

[3] Chan, W., Zhou, X., Wang, W. and Chua, T.S., (2012), July. Community answers summarization for multi-sentence question with group L 1 regularization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 582-591).

[4] Patel, C., Patel, A. and Patel, D., (2012), Optical character recognition by open source OCR tool tesseract: A case study. *International Journal of Computer Applications*, *55*(10), pp.50-56.

[5] Proceedings of the VLDB Endowment, Vol 7, 2013-2014 *http://www.vldb.org/pvldb/vol7.html*

[6] Salton, G., Singhal, A., Mitra, M. and Buckley, C., (1997), Automatic text structuring and summarization. *Information Processing & Management*,*33*(2), pp. 193-207.

[7] Chuang, W.T. and Yang, J., (2000), July. Extracting sentence segments for text summarization: a machine learning approach. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 152-159). ACM.

[8] Chen, J. and Chen, H., (2013), A structured information extraction algorithm for scientific papers based on feature rules learning. *Journal of Software*, *8*(1), pp.55-62.

[9] Gupta, V. and Lehal, G.S., (2010), A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, *2*(3), pp. 258-268.

[10] Prabhakar, M. and Chandra, N., (2012), Automatic Text Summarization Based On Pragmatic Analysis. *International Journal of Scientific and Research Publications*, *2*(5).

[11] Edmundson, H.P., (1969), New methods in automatic extracting. *Journal of the ACM (JACM), 16*(2), pp. 264-285.