Sentence Detection and Extraction in Machine Printed Imaged Document using Matching Technique

Shalini Puri
Department of Computer Science
Birla Institute of Technology
Ranchi, India
eng.shalinipuri30@gmail.com

Satya Prakash Singh
Department of Computer Science
Birla Institute of Technology
Ranchi, India
spsinghbit@yahoo.co.in

***Abstract*— Sentence extraction is a new, challenging and critical step in the printed scanned imaged documents. In this paper, an efficient 4-layered Sentence Detection and Extraction System (SDES) model is proposed which is designed to detect and extract sentences from machine printed imaged document. Its internal details and architecture clearly show that how it processes an image to find out the underlying sentences. The basic idea is to first preprocess the imaged document for noise removal and skew correction, and then textual entities are detected and segmented at page, line and word levels. Firstly, the horizontal and vertical projection profiles are taken to segment and separate the lines and words. After skew correction, two stage Character Based and Word Based Leveled matching and testing are performed, which verify and identify the correct character and word by searching for similar textual characters and words in Character Set Storage (CSS) and Word Pseudo Thesaurus (WPT). If any word pattern is not matched and identified by WPT, then it is stored in the Unmatched Word Storage (UWS) for the future reference. Such testing and verification are used at two levels to increase the accuracy% of SDES, and thereby, reducing the errors. It increases the system performance greatly. Finally, all the sentences of imaged document are extracted. Experimental results are found at the word, character and sentence levels. Their accuracy% results are good which show the high system performance and efficiency.**

Keywords*— *projection profile; line segmentation; word segmentation; character recognition; storage bins; sentence extraction; natural language processing

I. INTRODUCTION

Sentence extraction and processing has been a long time active area of research in document analysis and natural language processing. Many researchers have focused on sentence extraction and information retrieval to identify the underlying contents from text documents. Such research work analyzes the information and content retrieval from web pages, online available text documents, document files etc. A new, challenging and innovative research area in text retrieval is to detect, identify and locate the sentences in a machine printed imaged document [1] - [20]. Image processing field is used to process a given image to establish the viability of proposed solutions to a given problem [1] [3]. Thus it also includes the concept of pattern recognition for the purpose of correctly identifying the character patterns or character images.

In digital image processing, first step is the image acquisition. After this, the image is further analyzed by going through a series of sequential steps and operations to obtain the desired results. Document image analysis is a tedious and important part whose goal is to shape and transform the image into the required form, so that the resultant transformed image can be used for other purposes, say content extraction, information retrieval, translations etc. Major text extraction steps include the document preprocessing, page, line and word segmentation and finally character recognition. In the recent years, many good methodologies, techniques and algorithms of text extraction have been developed with great profound experimental results by the researchers for various Indian and Non-Indian languages. But most of the research work has been focused on English imaged documents for text and content extraction. The proposed work is also a contribution for English imaged document sentence analysis and detection.

Sentence extraction systems [10] – [14] automatically handle and process large scale text imaged documents to reduce efforts and time complexity. These systems not only separates text and non-text data but also have key applications in the fields like extraction of Meta-Information [10], natural language processing [10] – [15], text summarization [11] – [14], image content classification [16], paragraph extraction [18], image mining and many more.

The rest of the paper is organized as follows. Section II presents the related work to text extraction and content analysis. In the next section, proposed methodology and its detailed design and architecture are discussed. Section IV presents the experimental results performed on system and accuracy is calculated. Finally, section V presents the conclusion and future extensions.

II. RELATED WORK

Over the last few decades, many content and text extraction methods and techniques have been proposed in the literature. For content extraction from document image, Habibi et. al. presented an approach of keyword searching by extracting a feature vector with wavelet transform for sub-

978-1-4673-8253-3/15/$31.00 ©2015 IEEE

words which are further clustered with SVM [4], whereas an automatic feature discovery method was discussed by Wang et. al. [5] which locates and segments the regions in machine-printed and handwritten document images. For Content extraction, Baird et. al. also reported a methodology for training classifiers to extract document image contents by locating and segmenting the regions containing handwriting, machine-printed text, photographs, blank space, etc. [6].

Bukhari et. al. proposed a ridge-based method as a generic and robust text-line finding approach. It used filter bank smoothing followed by ridge detection without having any preprocessing or post-processing step [7]. A comprehensive overview on the state-of-the-art extraction methods for heterogeneous contents such as images and text was provided by the Bagadkar et. al. in [8].

Some researchers also focused on text extraction from printed documents which is available in different styles and formats. Hirano et. al. presented a method to extract text and layout information (character size, position and table position) from document files of various formats and graphic elements (text, image and path objects) by analyzing Page Description Language (PDL) generated from printed document [9], whereas Garain et. al. explained the concept of extraction and listing of important sentences from printed documents by finding out those titles, authors' names, subtitles, references and sentences which have their type styles as italic, bold and all capital. This helps in content summarization [10].

Bloomberg and Chen et. al. have done a lot of work in the direction of image document summarization. In [11], they presented a system for selecting sentences of imaged document to provide summary without using OCR. The sentences are selected based on a set of discrete features by characterizing the words within a sentence and the location of the sentence within the imaged document. Their another approach of sentence extraction without using OCR for summarization purpose is to first find text regions, text lines and words, and then to identify the sentence and paragraph boundaries. They determined the stop words by computing a set of word equivalence classes [12]. Another contribution [13] for document summarization from a scanned document without using OCR was based on text-related features extraction. In the same direction, they also demonstrated a font independent system to detect and locate partially specified keywords in scanned images using Hidden Markov Models which does not require pre-segmentation step [14].

Desilva et. al. [15] proposed an algorithm for the detection of proper nouns in document images, which were printed in mixed upper and lower case. So, analysis of graphical features of words in a running text was performed to determine words that may be the names of specific persons, places, or objects. It helps in those word recognition techniques where word images are matched to entries in a dictionary. To segment a page, Felhi et. al. proposed a hybrid approach based on connected component and region analysis by applying the stroke descriptor and contour model [16]. Cheriet et. al. discussed a stroke model to depict the local features of character objects as double-edges in a predefined size, and thereby, detected thin connected components selectively, but ignored large backgrounds [17]. The method proposed by Waked et. al, detected and corrected the skew of a document image, segmented the page into text and graphical components which were further segmented into paragraphs and lines and classified the script type [18]. Dhandra et. al. proposed a method of skew angle estimation in a binary document image based on image dilation and region labeling technique [19]. Jetley et. al. presented a binarization approach, in which after pre - segmentation, it used a fringe map based text line segmentation algorithm to define text context at the cost of increased processing time [20].

III. Sentence Detection and Extraction

Sentence Detection and Extraction System (SDES) model is designed to detect, identify and extract the sentences from a scanned imaged document as shown in Fig. 1. The aim of SDES is first to extract the text as characters, and then to words and finally the sentences.

Let D_T be a A4 sized text document and D_I be the imaged document containing only textual data. After that, a series of pre-processing steps are performed which includes noise removal, binarization of the image, skew detection and correction. The resultant image is further sent for detection and segmentation steps at Page, Line and Word levels to recognize each character of each word of each line of D_I. Then character - level and word - level matching and testing are performed to identify them correctly, so that their equivalent text forms are compared with characters stored in Character Set Storage (CSS) and with words stored in Word Pseudo Thesaurus (WPT) respectively, whereas undetermined or incorrect words are stored in Unmatched Word Storage (UWS). Finally, sentences are identified and extracted.

A. Proposed Methodology

The basic design and structure of the SDES model is shown in Fig. 1, which is characterized into four layers. In the first layer, image D_I is preprocessed to remove noise and outliers, and then binarized and skew corrected. Then the outer layout of the modified D_I is detected, which is further sent for the line and word level detection and segmentation.

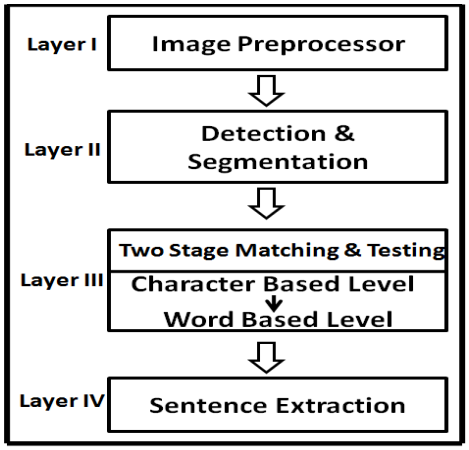

Fig. 1. A 4 – Layered Sentence Detection and Extraction System (SDES) Model.

978-1-4673-8253-3/15/$31.00 ©2015 IEEE

After performing these steps, each character is detected and identified, and then matched against CSS and its equivalent text form is stored in the character array. Similarly, each word is detected and identified and matched against WPT. If the word pattern lies in WPT, then it is stored in a string array, otherwise the word is rejected and stored in UWS. That rejected word may be a noun or any noisy word. If it is a noun or in some similar form, then it is used directly in sentence construction, else the noisy word needs to be corrected. Such noise may reduce the system performance and accuracy. The chances of such wrong interpretation are less, i.e., near 2 - 3 %, because the characters and words both are matched against their respective storages. As sentences are separated on the basis of three stop words or sentence terminators – a dot (.), a semicolon (;) and a question mark (?), so after detecting all the words, sentences are extracted in layer IV.

B. Detailed Design of SDES

In this subsection, the detailed design and architecture of basic SDES model are described systematically. Detailed SDES, shown in Fig. 2, is structured in six stages. First stage includes the image acquisition and its preprocessing. Stages II, III and IV are the corresponding parts of the detection sub-layer and segmentation sub-layer. These sub-layers determine the layouts of D_I (or page), line images and word images. It results in the identification and detection of a character image patterns. So, in the second stage, the structure layout of D_I is determined, and page is segmented using horizontal projection profile to detect and identify the line image patterns. In stage III, a line structure layout is detected using vertical projection profile to further detect and identify the word image patterns. In stage IV, the word structure layout is determined, so that a word is broken into its constituent parts, i. e., character image patterns, where each character pattern is made up of black pixels without having any gap between them. A character image pattern is called a Connected Component (CC) and a word pattern is a set of CCs.

After this, in stage V, character level and word level matching and testing with respective storage bins are performed. For this, three types of data storage bins are used as given below. First is the, Character Set Storage (CSS) which is a sort of alphanumeric character bin to store all alphabets, numerals and special symbols. Second is the Word Pseudo Thesaurus (WPT), a dictionary which is used to store all English words available. Third is the Unmatched Word Storage (UWS) which is storage to store the undetermined words which are not present in WPT. It can also include the names, nouns or such similar words. Such criterion helps in finding out the accuracy% and error% of the retrieved word from D_I.

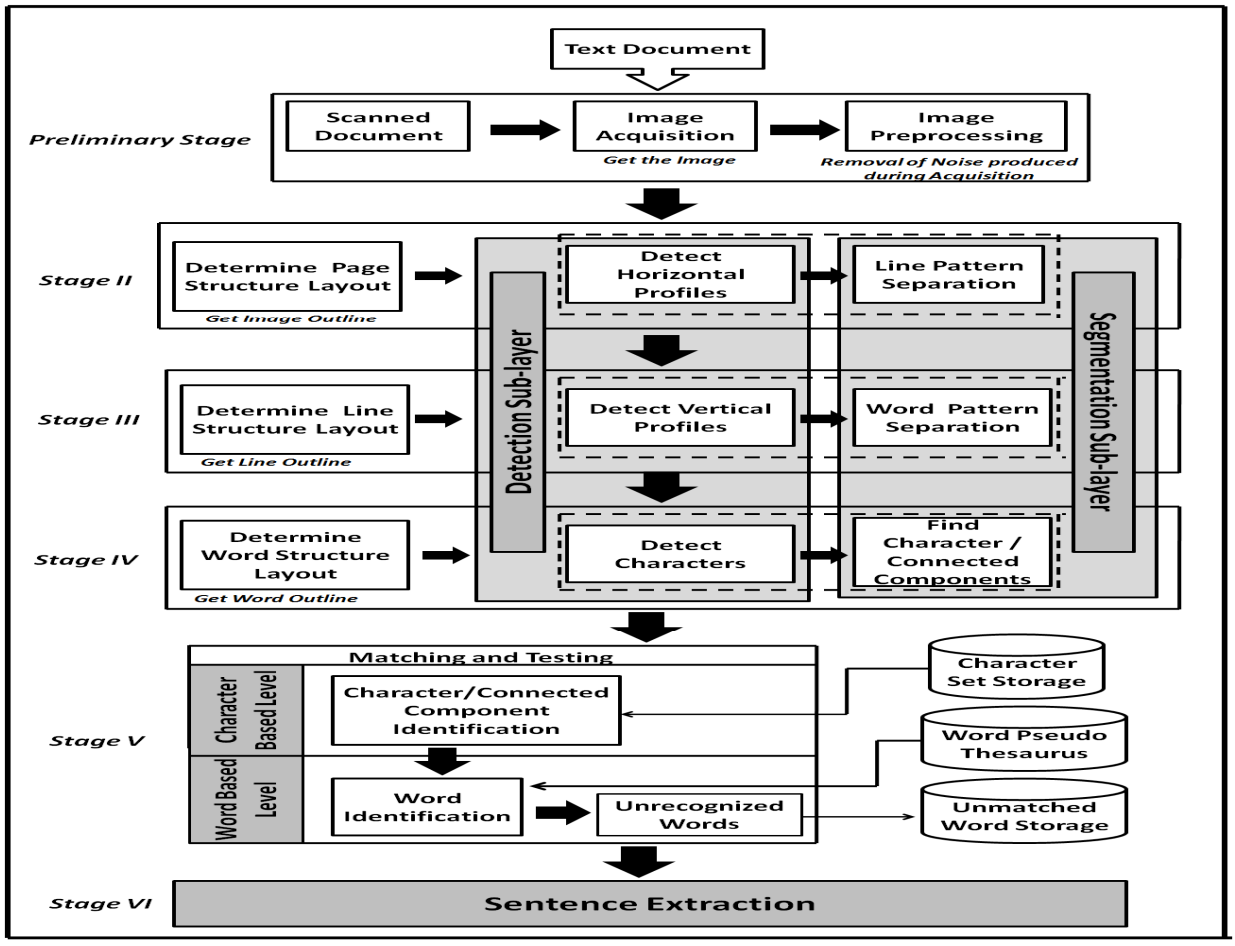

Fig. 2. Detailed Internal Design and Architecture of SDES Model.

978-1-4673-8253-3/15/$31.00 ©2015 IEEE

In stage VI, sentences are retrieved. All the text words (of equivalent word patterns) are collected and stored in Document Line Word called DLW [][], which are used to form the sentences. The criterion of sentence formation is to find out a sentence terminator – a dot (.), a semicolon (;) or a question mark (?). Various stages and their underlying key concerns are discussed below in more detail. differentiate among departments of the same organization). This template was designed for two affiliations.

1) Page Segmentation and Line Pattern Detection: In this step, after identifying the outer layout and boundaries of the page, horizontal projection profile of the image D_I is found. It is shown as the peaks and valleys in the profile, where a peak represents the long runs of black pixels and a valley represents the long runs of white pixels. Therefore, it results in separation of line patterns in D_I where total number of lines in D_I is stored in a Document Line Counter called DLC.

Algorithm Page_Seg_Line_Det ()

STEP 1: Initialize DLC as 0.

STEP 2: Determine horizontal profile of D_I.

a. Loop if a peak appears then there is a long run of black pixels.

b. DLC = DLC + 1.

c. Else there is a long run of white pixels.

d. Continue STEP 2 until no peak remains uncovered.

2) Line Segmentation and Word Pattern Detection: In this step, each line is segmented and word patterns are detected using vertical projection profile. Here, a Document Word Counter called DWC and a Line Word array LW [ ] are maintained which are used to store total number of word patterns of D_I and to store total number of word patterns occurred in each line pattern. Final step is to check that DWC and LW [ ] must be equal to each other.

Algorithm LineSeg_WordDet ()

STEP 1: Initialize DWC as 0.

STEP 2: Declare an array LW [ ] and assign 0 to LW [ ].

STEP 3: Determine vertical profile of each line of D_I.

a. for (i = 0; i <= DLC – 1; i++) // if a peak appears then there is a long run of Black Pixels.

i. LW [i] = LW [i] + 1.

ii. DWC = DWC + 1.

b. Continue STEP 3 until no peak remains uncovered.

STEP 4: Test and Verify as shown in equation (1) that

$$DWC = \sum_{i=0}^{DLC-1} LW\ [i] \qquad (1)$$

3) Word Pattern Segmentation, Character Pattern Recognition and Matching: As we know that each character is a connected component (CC) and a word is a set of CCs. So, to segment a word pattern, CCs are detected as attached long black pixels. For example, the image pattern of letter 'A' contains attached black pixels with no gap between them, so such long run is identified. For that, two arrays, a character array, i.e., Document Line Word Character called DLWC [j] [k] [l] and a Document Line Word string array called DLW [j] [k] are used. DLWC [j] [k] [l] is used to store the l^{th} character of k^{th} word of the j^{th} line, where $0 \le l \le CC_L - 1$, $0 \le k \le LW\ [j] - 1$, $0 \le j \le DLC - 1$ and CC_L depicts the last connected component (character pattern) of the word W_{jk}. DLW [j] [k] is used to store the k^{th} word of the j^{th} line, where $0 \le k \le LW\ [j] - 1$, $0 \le j \le DLC - 1$. In addition to them, a Document Sentence Counter (DSC) is used to count the total number of sentences present in D_I. A two stage character based and word based levels matching and testing is given below. Such testing is performed on two levels which drastically reduces the number of errors in character and word recognition and increases the accuracy % .

a) Character Based Level: At this substage, firstly a character image is identified, checked and then searched in CSS to find its equivalent text form alphabet CCjkl. If it is found, then that character is stored in DLWC [j] [k] [l]. While identifying the character patterns and searching their equivalent forms, if any sentence terminator – a dot (.), a semicolon (;) or a question mark (?) occurs and is identified, then it is a sentence, so DSC in incremented by one.

b) Word Based Level: Words are retrieved from DLWC [j] [k] [l] which are further matched and verified with WPT for correct word recognition. Some related concerns are discussed below.

- If a CC_L in a last word pattern DLC [j] [LW [j] – 1] [CC_L – 1] of a line pattern *j* contains a special character '-', where $0 \le j \le DLC - 2$, then it shows an incomplete word and must be continued at the first word pattern (0^{th}) of next line *(j + 1)* with the same special character. These two words must be merged into one by deleting both special characters; otherwise the word will not get matched in WPT. It further reduces the DWC by 1 and LW [j] will be updated.
- If last connected component CC_L, stored at DLWC [j] [k] [CC_L – 1] of a word pattern *k* of a line pattern *j* contains a special character ',', '!', '%' etc., then this character will be removed and discarded. Finally, the word is updated.
- If the last connected component CC_L, stored at DLWC [j] [k] [CC_L – 1] of a word pattern *k* of a line pattern *j* contains a special character '.', ';' or '?', then it is considered as the word terminator.

Algorithm WordSeg_CharDet ()

STEP 1: Declare a character array DLWC [j] [k] [l] and a string array DLW [j] [k].

978-1-4673-8253-3/15/$31.00 ©2015 IEEE

STEP 2: Initialize DSC = 0.

STEP 3: // Character Identification and Verification

a. for (j = 0; j ≤ DLC – 1; j++) {

b. for (k = 0; k ≤ LW [j] – 1; k++) {

Recognize the character as one CC_{jkl}.

Match CC_{jkl} with most probable character from CSS.

for (l = 0; l ≤ CC_L – 1; l++) {

Store the equivalent character of recognized character pattern in DLWC [j] [k] [l].

// Counting Total number of Sentences (Sentence Detection)

if DLWC [j] [k] [CC_l – 1] = '.' ';' or '?'

then DSC = DSC +1 }

STEP 4: // Word Matching and Identification

a. Store the word in DLW [j] [k].

b. Match DLW [j] [k] with WPT.

if W_{jk} lies in WPT, recognize W_{jk}.

else

{W_{jk} may be a noun or a wrong recognized word.

Store the word in UWS.} } }

STEP 5: // Accuracy Computations as shown in equations (2) and (3).

$$Accuracy\ \% = \frac{Total\ Number\ of\ Characters\ Identified\ Correctly}{Total\ Number\ of\ Characters} \quad (2)$$

$$Accuracy\ \% = \frac{Total\ Number\ of\ Words\ Identified\ Correctly}{Total\ Number\ of\ Words} \quad (3)$$

4) Sentence Extraction: Final stage of SDES model is the sentence extraction. DSC gives the total number of sentences which are present in DI. It gives a base for the occurrence of sentences in imaged document. So, a Document Sentence Storage (DS) is used which lists out all the extracted sentences, stored in a string array, DS [DSC]. The accuracy% of sentences extracted correctly is calculated as shown in equation 4.

$$Accuracy\ \% = \frac{Total\ Number\ of\ Sentences\ Identified\ Correctly}{Total\ Number\ of\ Sentences} \quad (4)$$

IV. Experimental Results

The proposed SDES model was tested on a variety of printed English documents. Documents are scanned using HP LaserJet M1005 MFP Printer with 300 DPI resolution to obtain the imaged document D_I. Experiments were performed on two types of imaged documents, first are the scanned images of the pages of a Book (B) and second contain the scanned images of .doc files (W). Total six document images were tested where two pages (B1 and B2) are scanned from a Book and other four are scanned images of .doc files (W1, W2, W3 and W4). Firstly, an image was converted into a gray scale image and then preprocessed. This system is capable to segment the page and line accurately using projection profiles. Fig. 3 shows the horizontal profiles obtained for document images of B2 and W4, where W4 was a colored image and projected efficiently by the system.

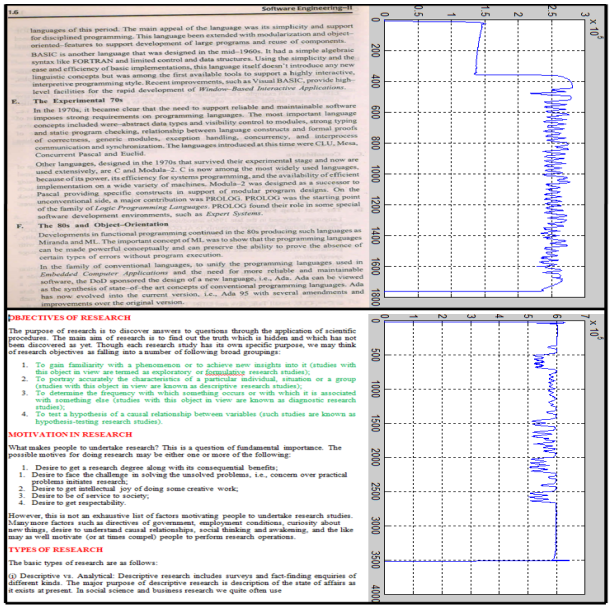

Fig. 3. Scanned Documents B2 and W4 and Their Corresponding Horizontal Projection Profiles.

Table I shows the obtained results of DLC, DWC and LW [] for all six documents. It can be clearly seen that DWC = LW [ ]. Finally, word accuracy is computed on the basis of number of unidentified and unmatched words. But the system fails to detect italic words correctly. In table II, computation of all characters and sentence extraction are shown with their obtained accuracy results. While finding out the sentences of B1, low accuracy results were obtained near about 65.62%. Reason behind such inaccuracy is that in two documents B2 and W4, last sentences of both were not terminated, so they remained identified. Second Reason is that if some sentence terminators are put randomly anywhere in the document, then it also leads to the misclassification of a sentence, and thereby reduces the sentence recognition accuracy. Still, the obtained results are encouraging.

TABLE I. Computing Document Lines, Words, Undentified and Unmatched Words with Word Accuracy Estimation

Document Number	DLC	DWC	LW[ ]	Unidentified Words	Unmatched Words	Word Accuracy %
B1	38	454	454 (= (18 + 13)*2 + 17 + 16 * 5 + (15 + 14)*6 + (12 + 3)*3 + 11*4 + 10 + 7 + 6 + 5 + 4 + 0)	20	1	95.37%
B2	38	421	421 (= 16 + (15 + 14) * 6 + (12 + 11) *5 + 13 * 4 + (10 + 7 + 5 + 4) * 2 + 9 + 3 + 0)	12	17	93.11%

978-1-4673-8253-3/15/$31.00 ©2015 IEEE

Document Number	DLC	DWC	LW[ ]	Unidentified Words	Unmatched Words	Word Accuracy %
W1	16	128	128 (= 15 + 14 + 13 + 12 + 11 + (10 + 6) * 2 + 9 + 7 + 5 + 4 + 3 + 2 + 1)	0	0	100%
W2	16	128	128 (= (15 + 11 + 10 + 6 + 5 + 1) * 2 + 13 + 3 + 14 + 2)	0	0	100%
W3	39	503	503 (= (18 + 13 + 11 + 6 + 5) * 2 + (17 + 16 + 12) * 4 + 15 * 6 + 14 * 7 + 10 + 8 + 7 + 4)	3	2	99%
W4	32	360	360 (= 19 + (16 + 15) * 4 + (14 + 12) * 6 + 8 * 2 + 3 * 5 + 13 + 11 + 5 + 1	0	0	100%

TABLE II. COMPUTING DOCUMENT CHARACTER AND SENTENCE ACCURACY

Document Number	Total Characters Identified	Total Characters Identified Correctly			Character Accuracy %	Total Sentences Identified Correctly	Sentence Accuracy %
		Alphabets and Digits	*Special Characters*	*Sentence Terminators*			
B1	2501	2312	33	32	95.04%	21	65.62%
B2	2460	2294	38	28	95.93%	21	75%
W1	648	640	0	8	100%	8	100%
W2	648	640	0	8	100%	8	100%
W3	2803	2720	38	38	99.75%	28	73.68%
W4	1930	1869	27	30	99.79%	20	66.66%

V. CONCLUSION AND FUTURE EXTENSIONS

The proposed SDES model shows a series of consecutive steps from image acquisition to sentence extraction. Preprocessing steps are used to normalize the image and character based level and word based level matching are used to increase system performance and accuracy %. Obtained results are promising. Some improvements are required to increase accuracy%. Future work and extensions are: first is the implementation of strong and improved algorithm for matching stage and second is implementation of classification algorithm for image mining using the concept of syntax and semantics of extracted sentences.

References

[1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed. Prentice Hall of India, 2007.

[2] E.Gose, Pattern Recognition and Image Analysis, 1st ed. Prentice Hall of India, 2009.

[3] S. Marinai, "Introduction to Document Analysis and Recognition," Machine Learning in Document Analysis and Recognition, Vol. 90, pp. 1-20, 2008.

[4] M. Habibi and R. Azmi, "Content Based Document Image Retrieval with Support Vectors Clustering," Proc. of International E-Conference on Information Technology and Applications, Vol. 2, Issue 5, pp. 308-314, 2012.

[5] S. Y. Wang, H. S. Baird and C. An, "Document Content Extraction Using Automatically Discovered Features," IEEE Tenth International Conference on Document Analysis and Recognition, pp. 1076-1080, 2009.

[6] C. An, H. S. Baird and P. Xiu, "Iterated Document Content Classification," IEEE Ninth International Conference on Document Analysis and Recognition, pp. 252-256, 2007.

[7] S. S. Bukhari, F. Shafait and T. M. Breuel, "Towards Generic Text-Line Extraction," IEEE Twelth International Conference on Document Analysis and Recognition, pp. 748-752, 2013.

[8] S. L. Bagadkar and L. G. Malik, "Review on Extraction Techniques for Images, Text Lines and Keywords from Document Images," IEEE International Conference on Computational and Computing Research, pp. 1-3, 2014.

[9] T. Hirano, Y. Okano, Y. Okada and F. Yoda, "Text and Layout Information Extraction from Document Files of Various Formats Based on the Analysis of Page Description Language," IEEE Ninth International Conference on Document Analysis and Recognition, pp. 262-266, 2007.

[10] U. Garain and B. B. Chaudhuri, "Extraction of Type Style Based Meta-Information from Imaged Documents," IEEE Proc. of Fifth International Conference on Document Analysis and Recognition, pp. 341-344, 1999.

[11] F. R. Chen and D. S. Bloomberg, "Extraction of Indicative Summary of Sentences from Imaged Documents," IEEE Proc. of Fourth International Conference on Document Analysis and Recognition, Vol.1, pp. 227-232, 1997.

[12] D. S. Bloomberg and F. R. Chen, "Document Image Summarization without OCR," IEEE Proc. of International Conference on Image Processing, Vol. 1, pp. 229-232, 1996.

[13] D. S. Bloomberg and F. R. Chen, "Extraction of Text-related Features for Condensing Image Documents," The International Society for Photo-Optical Instrumentation Engineering Document Recognition III, pp. 72–88, 1996.

[14] F. R. Chen, L. D. Wilcox and D. S. Bloomberg, "Detecting and Locating Partially Specified Keywords in Scanned Images using Hidden Markov Models," Proc. of Second International Conference on Document Analysis and Recognition, pp. 133-138, 1993.

[15] G. L. Desilva and J. J. Hull, "Proper Noun Detection in Document Images," Pattern Recognition, Elsevier Science Ltd, Vol. 27, No. 2, pp. 311-320, 1994.

[16] M. Felhi, S. Tabbone and M. V. O. Segovia, "Multiscale Stroke-Based Page Segmentation Approach," IEEE 11th IAPR International Workshop on Document Analysis Systems, pp. 6-10, 2014.

[17] X. Ye, M. Cheriet and C. Y. Suen, "Stroke-Model-Based Character Extraction from Gray-Level Document Images," IEEE Transactions On Image Processing, Vol. 10, No. 8, pp. 1152-1160, 2001.

[18] B. Waked, S. Bergler, C. Y. Suen and S. Khoury, "Skew Detection, Page Segmentation, and Script Classification of Printed Document Images," IEEE International Conference on Systems, Man and Cybernetics, Vol. 5, pp. 4470-4475, 1998.

[19] B. V. Dhandra, V. S. Malemath, H. Mallikarjun and R. Hegadi, "Skew Detection in Binary Image Documents Based on Image Dilation and Region labeling Approach," IEEE 18th International Conference on Pattern Recognition, Vol. 2, pp. 954-957, 2006.

[20] S. Jetley, S. Belhe, V. K. Koppula and A. Negi, "Two-Stage Hybrid Binarization around Fringe Map based Text Line Segmentation for Document Images," IEEE 21st International Conference on Pattern Recognition, pp. 343-346, 2012.

978-1-4673-8253-3/15/$31.00 ©2015 IEEE