Telugu Word Segmentation Using Fringe Maps

Koteswara Rao Devarapalli[1,2(✉)] and Atul Negi[2]

[1] Department of Computer Science and Engineering,
Mahatma Gandhi Institute of Technology, Hyderabad 500075, India
dkrao@mgit.ac.in
[2] School of Computer and Information Sciences, University of Hyderabad,
Gachibowli, Hyderabad 500046, India
atulcs@uohyd.ernet.in

Abstract. In this paper, we propose a word segmentation method that is based on fringe maps on Telugu script. Our objective is to create a data set of word images for enabling direct training for recognition on those. The standard methods employed for the task of word segmentation in Telugu OCR systems are projection profiles and run-length smearing. However those methods have their limitations. In this work a different application of fringe maps is shown for line segmentation into words. Fringes were previously applied successfully for carrying out classification and line segmentation. Telugu script, which has consonant modifiers that are usually placed below or below-right to the base consonants. This kind of orthographic property leads to characters that may touch each other. One way to deal with touched characters is to make use of segmentation free methods, which do not need prior segmentation of word images into characters or connected components. The novelty of our method is that we analyze fringe maps of document images to find an appropriate fringe value threshold and apply it for word segmentation of Telugu documents. Encouraging results are observed with our fringe value threshold based word segmentation. We observe that choosing higher threshold fringe values leads to under-segmentation of words, whereas lower values cause over-segmentation of words. Our word segmentation approach is successfully compared with the widely used projection profiles based word segmentation method.

Keywords: Akshara · Fringe distance · Telugu OCR · Word segmentation

1 Introduction

The current trend in Telugu optical character recognition (OCR) is that deep neural networks have been successfully demonstrated for improving the performance of existing systems. Compared to the traditional recognition approaches, deep learning frameworks demand significantly large training sets for effective

S. Sundaram and G. Harit (Eds.): DAR 2018, CCIS 1020, pp. 87–96, 2019.
https://doi.org/10.1007/978-981-13-9361-7_8

modeling. Now a significant percentage of efforts is required to prepare such training data.

The standard training data sets are not available in sufficient magnitude to develop robust, end-to-end Telugu OCR systems based on deep architectures [1]. Previous methods rely on either own data sets [7] or the standard corpus of 5000 document images scanned from the popular old books as the part of a consortium project. Due to practical difficulties of clarity of the images [6] cited results only on 1000 pages. Few authors prepared synthetic data sets and made them public for the research needs. Broken and touched characters are the major problems that can effect the performance of Telugu OCR systems. In our work we do not rely on segmentation of connected components or characters [2,10], but we segment words using fringe distance method. The goal of avoiding character segmentation is to prepare data set of words for enabling the recognition of broken and touching characters.

1.1 Properties of Telugu Orthography

Telugu has its own phonetic script, with well rounded *aksharas* (characters). Telugu text is composed of *aksharas*, which are the basic units of orthography and are made up of rounded curves. Telugu script consists of glyphs for basic vowels, basic consonants, vowel modifiers and consonant modifiers.

In Telugu script, there are 15 vowels and 35 consonants. It has corresponding vowel modifiers and consonant modifiers. Unlike Roman script, Telugu syllables have a direct correspondence to their orthographic units [8]. Vowel modifiers are placed at the top portions of the basic glyphs, whereas the consonant modifiers are found below or below-right places of the symbols. These modifiers and printing methods may cause broken and touched characters. The prevailing issues in Telugu character recognition are mainly due to broken and touched characters. The widely used conventional connected component based methods may fail due to the segmentation errors that may occur because of the broken and touched characters. In the following sections of the paper, the related work is reported in Sect. 2 and we describe our word segmentation approach in Sect. 3. The data preparation task is explained in Sect. 4, and finally the conclusion is made in Sect. 5.

2 Related Work

Our work mainly consists of data preparation through applying a novel fringe distance method to perform line and word segmentations. We avoid segmentation of either connected components or characters, because character level segmentation may cause broken and touched characters.

Fringe distance method involves generation of fringe map for a given binary document image. A fringe map is the function of fringe distance represented as numeric value per pixel, which is incremented on each move away from a foreground pixel. The concept of fringe distance to generate fringe maps of characters

is first employed for demonstrating character recognition [3]. It is also applied successfully for Telugu character recognition [8]. Consecutive Telugu text lines may overlap at few of the symbol positions due to some of the font types and glyphs for vowel modifiers and consonant modifiers. The standard methods of line segmentation such as horizontal projection, and run length smearing may fail and can cause segmentation error. The peak fringe number (PFN) concept is used to segment lines of Telugu document images with some overlap between them due to modifiers [4]. They segment lines of a document depending on fringe map creation, retaining peak fringe numbers between consecutive black pixels per each column, and filtering unwanted fringe numbers that occur inside characters.

In the recent work on Telugu character recognition [1], they prepared a synthetic data set for training Convolutional neural network classifier, which does not need any explicit feature extraction. It relies on connected component extraction, which causes segmentation errors. *Indic* scripts have got unique orthographic properties due to the existence of separate glyphs for vowel and consonant modifiers. One of the orthographic properties of Telugu script is that consonant modifiers are spatially spread across the middle and lower zones. The concept of peak fringe numbers (PFNs) is used for finding zones and to define a first level classifier before training models [9]. In their work, peak fringe numbers that occur due to white space inside characters is used for identifying middle line across characters. They employ the middle line as the reference in finding zones, which enable grouping of words into words with consonant modifiers and without consonant modifiers. This kind of grouping is aimed for effective character modeling. For word segmentation, a novel fringe distance based method is employed. Fringe maps of given binary document images can be computed using fringe distance method, which is previously used for classification [3], feature extraction [8] and line segmentation [4].

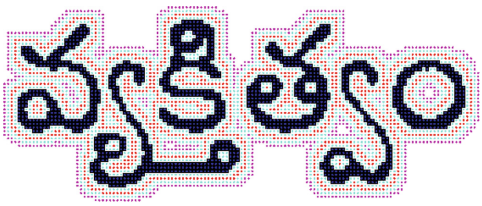

Fig. 1. Example fringe map of a word image pattern is shown. Fringe numbers are colored as 0-Blue, 1-Cyan, 2-Red, 3-Cyan, and 4-Magenta respectively. (Color figure online)

3 Word Segmentation

Projection profiles based word segmentation algorithm makes use of zeros in the projection that defines segmentation points. Run-length smearing algorithm (RLSA) can be used to perform marking in horizontal direction on the complement of input binary image for finding the segmentation points. In RLSA, we

change adjacent 0's to 1's if their count is less than or equal to a threshold to make smearing.

In this paper, we use the fringe maps of document images to attempt the word segmentation task for Telugu OCR. A fringe map gives various levels of the fringe (background) of a pattern. Different levels can be used for different segmentation needs such as line segmentation, word segmentation and character segmentation. Fringe map of an example word image is shown in Fig. 1.

3.1 Fringe Map

Fringes may be thought of as a kind of distance transform on background pixels in the image. Binarized text images are used for creating fringe maps of document images. Fringe map of a binary image can be computed through finding fringe distances. For an image, fringe distance of a pixel can be defined as the pixel distance to the closest foreground pixel. Fringe map was introduced for character recognition [3] and first used for Telugu OCR [8]. Towards improving Telugu OCR system, fringe maps were used for line segmentation [4] and for finding modifier zones of word images to enable classification of them into major classes such as words having modifiers and words having no modifiers [9].

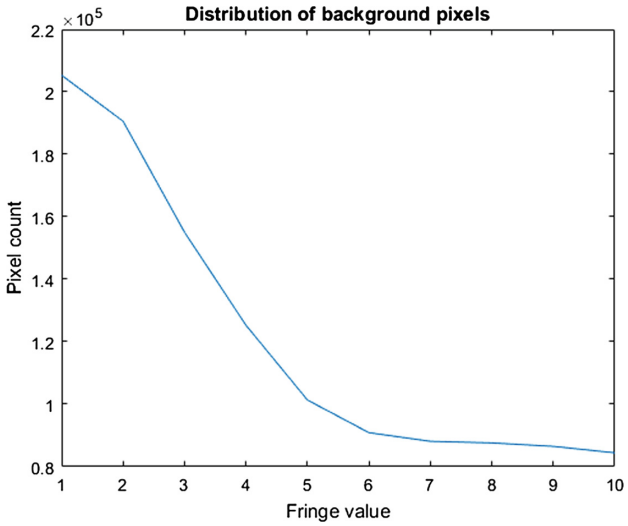

Fig. 2. Distribution of fringe values of a document image. The higher pixel counts are due to fringe values such as 1, and 2. Whereas lower counts are due to fringes that occur as we move away from the foreground text.

The first step in fringe map creation is to assign 0 and −1 fringe values to every black and white pixels respectively. Then we look for white pixels in the horizontal, vertical, and diagonal directions of every black pixel and are set to the

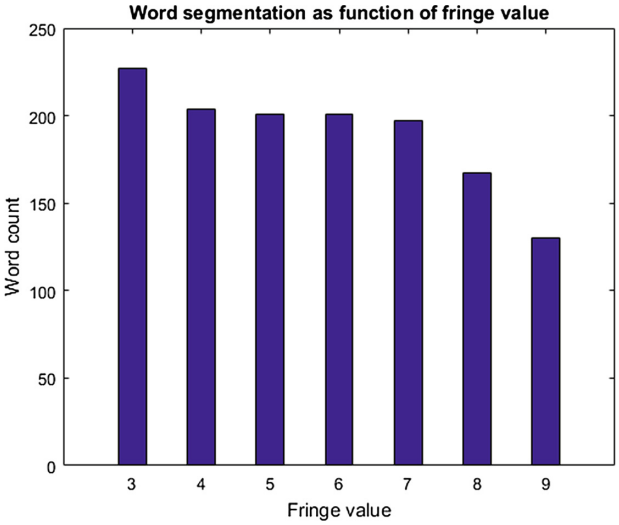

Fig. 3. Word count statistics for an example document image, over-segmentation of words is observed for the fringe values less than or equal to 3, whereas under-segmentation is found for the fringe values greater than or equal to 6.

fringe 1. Further pixels with fringe value 1 are followed to label their neighbors having -1 with fringe 2. This numbering is continued until there are no pixels with number -1. Thus, for each input document image, we compute its fringe map.

3.2 Fringe Value Thresholding Algorithm

The goal of this work is to segment word regions from document images for Telugu OCR. We use the standard preprocessing methods such as Otsu's binarization to convert Gray scale images into binary ones and median filter with 3×3 kernel to remove commonly occurring noise. The proposed Telugu word segmentation involves different tasks on input document images. Input data to our word segmentation method can be either document images or line images. The main tasks involved are creating fringe map, filtering unwanted fringe values, filtering smaller undesirable connected components, locating word regions, finding segmentation points, and extracting word images. Fringe maps were used [3] for character recognition. The novelty of our work is that fringe maps are filtered based on a fringe value threshold, which is chosen empirically to facilitate the finding of word segmentation points.

We analyze the distribution of fringe values of different document images. Distribution of fringes for a document image is shown in Fig. 2. The higher pixel counts are due to fringe values such as 1, and 2. Whereas lower counts are due to fringes that occur as we move away from the foreground text. The purpose this analysis is to know the distribution of background pixels around foreground

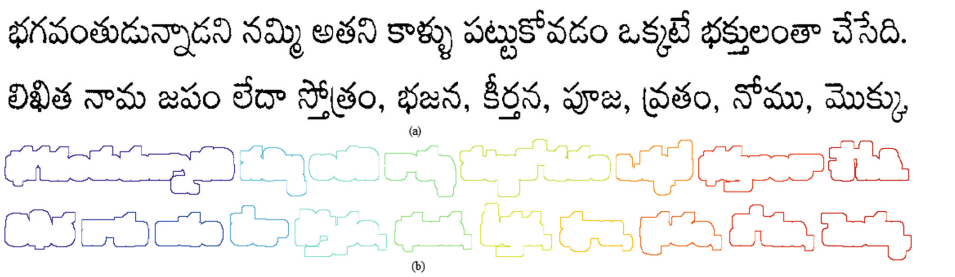

Fig. 4. Word Segmentation: (a) Normal document image, and (b) Word regions having fringe value 5.

text. It facilitates us to choose a threshold fringe value. The segmentation of words as the function of fringe value is shown in Fig. 3. We have chosen 5 as the threshold fringe value for performing word segmentation, because we observe under-segmentation above this threshold. A portion of Telugu document image and corresponding labeled word regions using fringe value threshold 5 is shown in Fig. 4. For a range of 1–10 fringe values, we have empirically observed the distribution of fringes around the text. Towards the task of filtering unwanted fringes, we have chosen fringe value 5 as the threshold(th1). Filtering involves keeping the pixels with threshold and setting values of all other pixels to zero. We also filter out smaller undesirable connected components, which are having

Algorithm 1. Word Segmentation

procedure WordSegmentation(*DocImage*) ▷ DocImage or line image
  for all *whitePixels* **do** ▷ Create fringe map
    $whitePixel \leftarrow -1$
  end for
  for all *blackPixels* **do**
    $whiteNeighborsOfblackPixel \leftarrow 1$
  end for
  $label \leftarrow 1$
  repeat
    for all *Pixels with label* **do**
      $label \leftarrow label + 1$
      $whiteNeighborsOfPixels \leftarrow label$
    end for
  until *all whitePixels are labeled*
  ▷ Find word regions
  Filter unwanted fringe values using a threshold $th1$
  Filter unwanted smaller components having pixels less than a threshold $th2$
  Locate word regions
  Find segmentation points
  Extract word images
end procedure

Table 1. Performance of different word segmentation approaches for the segmentation of Telugu word images

Book	Pages	Text-words	Algorithm	Word-imgs	Errors
DiavamVaipu	122	21775	Fringe value thresholding	21832	57
			Projection profiles	21788	13
7-different books	30	5625	Fringe value thresholding	5622	15
			Projection profiles	5629	4

a count of pixels less than a threshold(th2), the value of th2 is 100. The goal of this task is to find the background regions around the words of text and these regions helps us to determine the segmentation points. The extraction of words is carried out relying on segmentation points. We compare our method with the widely used projection profiles based word segmentation approach. The results are given in Table 1. The projection profiles based approach causes over-segmentation when there is more gap between characters of words. The merit of this approach is that it can be applied on either document images or line images. We observe that higher threshold fringe values cause under-segmentation (merged words) shown in Fig. 5, whereas lower values lead to over-segmentation (split words). In this paper, we directly applied fringe value thresholding on document images. When there is no significant gap between lines, our method requires to execute on line images for extracting word images.

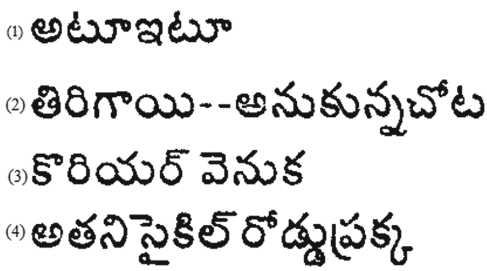

Fig. 5. Four cases are shown, each case has multiple words, which are segmented as single word due to merging of words.

4 Data Preparation

Data preparation involves segmentation of words from both standard corpus of document images and synthetic documents. Our Telugu standard corpus consists

of around 5000 document images of 26 Telugu books printed in different font type and size combinations. These document images are scanned with 300 dots per inch (DPI) from old popular books and the corresponding ground truth is created as part of the consortium projects.

Corpus also consists of degraded documents due to aging and poor print quality. In the previous works [5,6] 1000 document images of the standard corpus are used as the common bench mark data set. Further, we include some synthetic document images created using recently released Telugu Unicode fonts. The synthetic documents are scanned in different DPIs such as 200, 250 and 300. Deep learning methods can automatically learn features from the raw data and enable developing recognition systems with no explicit feature extraction. But, implicit learning features demands a lot of training data, particularly for the problems involving high-dimensional data such as images, videos.

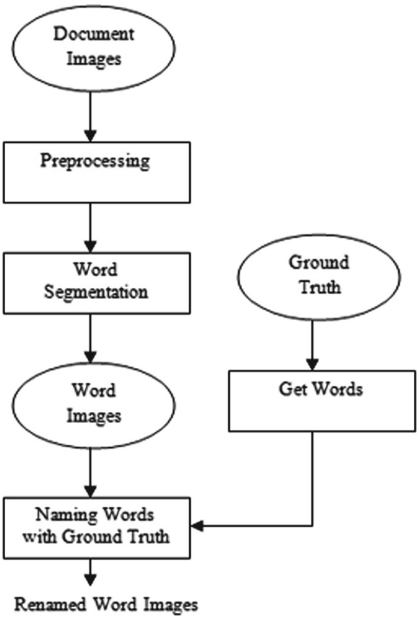

Fig. 6. Block diagram for data preparation.

Data preparation task consumes significant amount of time and can be expensive, especially when real world data is to be included. The recent trend in computer vision is that pre-trained models are available, which can be used to build sophisticated vision models using very little amount of data. Input to our data preparation task is the Telugu document images and their ground truth files. Output is the word images renamed with the corresponding ground truth. The sub-tasks of our data preparation are shown in Fig. 6. Name of each word is composed of numeric codes of the book, line, and word, as well as its ground truth,

which is a sequence of Telugu characters written in English script. The benefit of including ground truth in renaming word images is to enable straightforward training without requiring explicit label files. It also facilitates manual, and programmatic checking of the word images and their ground truth for correctness.

Data preparation involves two main sub-tasks: word segmentation and text processing. The former one consists of preprocessing the Telugu document images followed by segmentation of words. After the word segmentation is accomplished, the later main sub-task is Telugu text processing that involves isolation of ground truth words. For each ground truth text file, it is divided into lines and then words are isolated from lines of text. We rename each word image with its book code, line number, word number and ground truth. This kind of renaming enables easy verification of the word images against their ground truth for correctness.

4.1 Data Augmentation

Data augmentation is a method to increase our training samples to a large amount. Since deep learning methods demand big data for training the models, we apply augmentation methods to increase our word samples that consists of broken and touched characters.

The input to our data augmentation task is the word images extracted from the standard corpus of documents, which are scanned with the efforts of consortium project teams. Our main goal is to improve the Telugu OCR system performance despite the presence of broken and touched characters. We augment them through a number of random transformations, thus our model would never see twice exact the same image. This is to prevent over fitting and better generalize the model. It is particularly useful, when the training data set is small. We employ seven primary data augmentations for our data preparation: rotation, shift, rescale, shear, and zoom. For rotation, it is to rotate images in degrees (0–10), and shift transformation aims to translate images both vertically and horizontally. For rescale, we multiply the data by a value. Shear operation is for applying shearing transformations, whereas zoom is for zooming inside images. Further, fill mode is the strategy used for filling in newly created pixels, which can appear after a rotation or a shift. Thus data augmentation enables us to increase word samples that have a few instances in the corpus.

5 Conclusion

Our work makes use of the concept of fringe distance transform, which is used to generate fringe maps of Telugu document images. The word segmentation is achieved by choosing a threshold fringe value through analyzing the distribution of fringes of document images. The word segmentation is described as the function of fringe value. The merit of this approach is that it can be applied on either document images or line images. We observe that choosing higher fringe

values cause under-segmentation of words, whereas lower values lead to over-segmentation of words. In this paper, we directly applied fringe value thresholding on document images. When there is no significant gap between lines, our method requires to execute on line images to extract word images.

The data preparation involves extraction of word images from standard corpus, and then the application of augmentation methods to increase the word samples that involve instances of broken and touching characters. Further, we use Unicode based text processing to obtain ground truth words and rename the word images with their ground truth. Renaming facilitates easy verification of word images and their ground truth for correctness. This kind of data set is to be used for deep learning methods, which require significantly large data sets for better modeling.

References

1. Achanta, R., Hastie, T.: Telugu OCR framework using deep learning. CoRR abs/1509.05962 (2015)
2. Bhagvati, C., Ravi, T., Kumar, S.M., Negi, A.: On developing high accuracy OCR systems for Telugu and other Indian scripts. In: 2002 Proceedings of the Language Engineering Conference, pp. 18–23, December 2002
3. Brown, R.L.: The fringe distance measure: an easily calculated image distance measure with recognition results comparable to Gaussian blurring. IEEE Trans. Syst. Man Cybern. **24**(1), 111–115 (1994)
4. Koppula, V.K., Negi, A.: Fringe map based text line segmentation of printed Telugu document images. In: 2011 International Conference on Document Analysis and Recognition, pp. 1294–1298, September 2011
5. Krishnan, P., Sankaran, N., Singh, A.K., Jawahar, C.V.: Towards a robust OCR system for Indic scripts. In: 2014 11th IAPR International Workshop on Document Analysis Systems (DAS), pp. 141–145, April 2014
6. Kumar, P.P., Bhagvati, C., Negi, A., Agarwal, A., Deekshatulu, B.L.: Towards improving the accuracy of Telugu OCR systems. In: ICDAR, pp. 910–914. IEEE Computer Society (2011)
7. Lakshmi, C.V., Patvardhan, C.: An optical character recognition system for printed Telugu text. Pattern Anal. Appl. **7**(2), 190–204 (2004)
8. Negi, A., Bhagvati, C., Krishna, B.: An OCR system for Telugu. In: ICDAR, pp. 1110–1114. IEEE Computer Society (2001)
9. Rao, D.K., Negi, A.: Orthographic properties based Telugu text recognition using hidden Markov models. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 05, pp. 45–50, November 2017
10. Vasantha Lakshmi, C., Patvardhan, C.: A multi-font OCR system for printed Telugu text. In: 2002 Proceedings of the Language Engineering Conference, pp. 7–17, December 2002