Recognition Strategies for General Handwritten Text Documents

M. Shridhar, G. F. Houle, F. Kimura
University of Michigan-Dearborn, MI 48128, USA,
Kappa Image LLC, Oakland, CA 94612, USA,
Mie University, Tsu, Japan

Abstract—This paper presents document recognition strategies for an important application: Recognition of text document containing multiple lines of text data. A project to study the feasibility of recognizing essays written by middle school students is the focus of the second study. In this project, a scanned document is processed to extract individual lines of text from the essay, extract individual words from the line and then apply word recognition techniques to the extracted words. While individual lines of data are extracted accurately using gap information between lines, extraction of words is a much bigger challenge. Since the essays are written by middle school children, word boundaries are ambiguous, especially when words are written in a non-cursive discrete style. In these cases the gaps between words are sometimes smaller than the gaps between characters of the word causing errors in estimating the location of word boundaries. In this paper, we propose two classes of word boundaries: 1) strong boundaries due to large gaps between words, 2) weak boundaries due to small gaps between words. There are also cases when two words do not have a clear gap between them, but are rather joined to give the appearance of a single word. Results obtained from our Phase 1 study will be presented in the paper.

I. INTRODUCTION

Handwriting recognition has reached a stage, where a number of practical applications have emerged and are in commercial use. These include address recognition on mail pieces, recognition of all fields in bank checks, tax forms, coupons etc. However, many of these applications that have been successfully employed depend on high degree of correlation among the various components of the document. For example, in the reading of addresses on a mail piece, initial effort is directed toward recognition of the postal code, which also happens to be the easiest one to recognize using traditional OCR techniques. Once this is accomplished, the recognized postal codes (an ordered list of top choices) are used to generate a list of states (provinces), cities and a lexicon of potential street names. Since addresses are written in a structured manner (beginning with the addressee, followed by street number and name, followed by city, state and postal code), recognition can be applied in a sequential manner. The sequence is typically postal code, state name, city name, street number, and street name. Only in very rare instances, addressee recognition is attempted.

Similarly a bank check has several strongly correlated fields. These include the courtesy amount and the legal amount, account number, approved signatures and a list of potential payees for the specific account. Also words associated with the legal amount are restricted to a small subset of the general lexicon. The key challenges are accurate extraction of the different fields followed by accurate recognition of the data in those fields. Many commercial banks have deployed automatic check processing with great success.

While some successful applications have been deployed, recognition of a general handwritten text document remains a challenge. The primary difficulty associated with recognition of text document comes from the fact that documents contain complete sentences replete with all punctuations and other delimiters. Depending on who writes a document, there are several challenges including

- Extraction of lines from the document
- Extraction of words
- Recognition of words including words which are misspelled
- Contextual analysis on whether word pairs or word triplets are grammatically aligned

In general, handwritten text documents are very difficult to recognize without application of linguistic constraints. One would require use of word bigrams and trigrams to reduce recognition errors. In this paper, the authors focus on extraction of lines of text from a general handwritten document, extraction of words from the line, followed by word recognition that generates multiple choices so linguistic rules can be applied in the final stage. Figure 1 shows a typical text document generated by a middle school student.

A quick scan of the document gives the impression that the quality of the writing is very good and that it is easy to read and interpret. A subset of the document is shown in figure 2. This figure reveals many artifacts and challenges. Since every line is underlined, these need to be removed without chopping the characters into smaller pieces. Second, the spacing between words is very inconsistent and often smaller than the
My Favorite Animal

My favorite animal is the most simple of all creatures. It is a horse. I love these creatures so much, and for many reasons. The first one is because they are so pretty when they gallop around. It makes me wonder if they are ever sad, the way they run around. I also love horses because they help people in so many ways. The wonderful animals make sad people happy, they make people who are sick feel well and they make great pets. (If you have enough room.)

Horses also helped people long ago. Native Americans rode them around all the time. Sometimes the pioneers would ride them too.

These beautiful horses inspire me to run out and be free. I let myself forget all my worries. Sometimes I wonder if they even have worries. Horses just are always there to make you smile. I do wish I had a horse of my own.

Figure 1. A handwritten essay written by a middle school student

Figure 2. A subset of the document of figure 1

Figure 3. Pixel count in each row of text document

Figure 4. First three lines extracted from document shown in figure 1

Figure 5. Elimination of underlines and remnants from lines above and below

However, in extracting a line of text, it is important that we do not cut characters especially those that are characterized by ascenders (‘b’, ‘d’, ‘f’, ‘h’, ‘l’ etc.) and descenders (‘f’, ‘g’, ‘q’, ‘y’ etc.).

II. LINE EXTRACTION

The first task in recognition of a handwritten text document is the extraction of individual lines. An effective strategy that the authors utilized in this study utilizes the pixel count distribution across rows of the document. Assuming that the document is not skewed, pixel count shows a minimum between the lines and this effectively signals the end of a line of text. Figure 3 shows the pixel count for the document shown in figure 1. It is clearly seen that between the lines pixel count in a row is typically small. Care should be taken to avoid errors due to a single word in a line.
III. EXTRACTION OF WORDS FROM LINES

Word extraction is based on the assumption that the gaps between words are usually significantly larger than the gaps between characters in a word. While this works well in documents written in a purely cursive style, difficulties arise when dealing with documents written in a discrete of a combination of cursive/discrete styles. In such cases, gaps between characters are often larger than the gaps between words. This can be clearly seen in the document shown in figure 1. To overcome this difficulty, the authors propose strong and weak boundaries. Weak boundaries may or may not be actual word boundaries. This ambiguity is resolved at the recognition stage. Figure 6 shows word boundaries for a typical line extracted from a text document. Strong boundaries are marked by solid lines, while weak boundaries are shown as dotted lines. The authors also use detection of period ('.') and comma (',') to improve word boundary estimation. One consequence of artifact removal is the loss of the dot over the character 'i' and sometimes periods.

IV. RECOGNITION OF EXTRACTED WORDS

A lexicon directed algorithm is used. The number of the boxes (or segments) obtained by the disjoint box segmentation is generally greater than the number of characters in the word. In order to merge these segments into characters and find the optimal character segmentation, dynamic programming (DP) is applied using the total likelihood of characters as the objective function. The likelihood of each character is given by a discriminant function described further. To apply DP, the boxes are sorted left to right according to the location of their centroids. If two or more boxes have the same x coordinates at the centroids, they are sorted top to bottom. Numbers that are displayed above or below the boxes in figure 7 show the order of the sorted boxes. It is worth observing that the disjoint box segmentation and the box sorting process reduce the segmentation problem to a simple Markov process, in most cases.

For example, boxes 3 and 4 correspond to letter "v" of Seventeen, box 5 to "e", box 6 to "n" ... and so on. These assignments of boxes to letters are represented, for example, by

\[ L = \sum_{i=1}^{n} l(i, j(i-1) + 1, j(i)) \]

where \( l(i, j(i-1) + 1, j(i)) \) is the likelihood for \( i \)-th letter.

In the lexicon directed algorithm, an ASCII lexicon of possible words is provided and the optimal character segmentation is found for each lexicon word. All lexicon words are then ranked according to their optimal likelihood per character \( L(n/n) \) to select the best candidate word. The optimal assignment (the optimal segmentation) which maximizes the total likelihood is found by applying the dynamic programming technique. Character likelihood is calculated using a modified quadratic discriminant function (MQDF) proposed by Kimura et al.

V. CONCLUSIONS

The authors have developed an accurate line extraction scheme to extract individual lines of text data.

Using the new concept of strong and weak word boundaries, the authors have shown that all words are correctly analyzed and recognized. Using word recognition confidences, ambiguous word boundaries are resolved.

REFERENCES


