2011 International Conference on Document Analysis and Recognition

A Fast Alignment Scheme for Automatic OCR Evaluation of Books

Ismet Zeki Yalniz, R. Manmatha
Multimedia Indexing and Retrieval Group
Dept. of Computer Science, University of Massachusetts
Amherst, MA, USA, 01003
{*zeki,manmatha*}*@cs.umass.edu*

***Abstract*—This paper aims to evaluate the accuracy of optical character recognition (OCR) systems on real scanned books. The ground truth e-texts are obtained from the Project Gutenberg website and aligned with their corresponding OCR output using a fast recursive text alignment scheme (RETAS). First, unique words in the vocabulary of the book are aligned with unique words in the OCR output. This process is recursively applied to each text segment in between matching unique words until the text segments become very small. In the final stage, an edit distance based alignment algorithm is used to align these short chunks of texts to generate the final alignment. The proposed approach effectively segments the alignment problem into small subproblems which in turn yields dramatic time savings even when there are large pieces of inserted or deleted text and the OCR accuracy is poor. This approach is used to evaluate the OCR accuracy of real scanned books in English, French, German and Spanish.**

***Keywords*-OCR evaluation; sequence alignment; digital libraries**

I. Introduction

The aim of this paper is to evaluate optical character recognition (OCR) accuracy on a set of books and to do this for multiple languages. OCR evaluation usually requires knowledge of the ground truth. One can build the ground truth manually but it is a labor intensive task [1]. A common approach to obtaining ground truth is to typeset data, print and scan it and then run an OCR [2], [3]. Results can be compared to the known ground truth. Another approach involves generating synthetic data and adding noise using degradation models [4] and using that to estimate errors. Both approaches do not provide a good estimate of the error rates of OCR models for large book scanning projects (Google Books or the Internet Archive) because of the variety of errors. Old books have different fonts, the scanning process introduces blur, characters and words at the edge of a book are often warped and there are numerous other possible sources of noise. For example, the OCR error rate on Latin books from the Internet Archive is substantially higher than that for English books although both essentially use the same character set (with small differences) and the OCR engine is supposed to be able to recognize documents in both languages. This cannot be inferred using the two approaches above. What we, therefore, need is a true estimate of actual errors in books.

Feng and Manmatha [5] proposed the use of ground truth texts from the Project Gutenberg website [6] to estimate OCR errors. These public domain books have been proofread by volunteers and are, therefore, mostly free of OCR and transcription errors. One issue with these books is that formatting information (line, page breaks) has been removed so that we are essentially left with one long string of possibly half a million characters. They, therefore, approached the problem as one of aligning the OCR output with the Gutenberg version of the text using a Hidden Markov Model (HMM). A typical book treated as a string may easily have 500,000 characters. At this scale well-known string alignment techniques (for ex. [7]) are not applicable because of their computational and/or spatial complexity. Feng and Manmatha proposed the use of unique words in the vocabulary of the book as anchor points to segment long texts into shorter ones. Essentially, matching unique words from the two books are taken to be anchor points. To ensure robustness anchor matches are confirmed by verifying that n-grams around the anchor also match. The resulting segments are later aligned individually using a HMM based model and the aligned segments are concatenated according to their original order. This approach effectively scales the whole alignment problem into a number of manageable size problems that require far less computation and memory space. The technique works because many unique words in the ground truth are correctly recognized in spite of OCR errors [1]. Boschetti et al. [8] use the same approach to align multiple books for OCR error correction.

One issue with Feng and Manmatha's approach is that in some situations the stretch between two anchor words may be relatively large making the dynamic programming somewhat expensive. A second issue is that the HMM requires some probabilities to be estimated by training. Potentially, this could change with language and OCR (though their technique seemed to be relatively stable to the OCR used).

We, therefore, propose a new fast recursive alignment approach to estimating OCR accuracy for books. First unique words in the vocabulary of the book are identified. The unique words from the ground truth and the OCR output

[1]By Zipf's law, half of the vocabulary words in an English document are unique.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.157

are aligned using a longest common subsequence(LCS) algorithm. The texts are then anchored at the unique words and the text segmented in to smaller subsequences (i.e. each subsequence is the piece of text between two anchor points). Each subsequence may now be thought of as a document. If we look at the vocabulary of each subsequence (not the vocabulary of the book) we will now find words which are unique in it. Thus, each subsequence can now be aligned using its own set of unique words. The process is recursively repeated until the subsequences become very small and can then be aligned using an edit distance based string alignment algorithm. This approach is very fast and can be used to align a typical book with its ground truth in about 1 second using today's desktop computer.

This technique only estimates OCR accuracy for texts which have a ground truth in the form of a Gutenberg text or similar text. This is still useful for a variety of reasons. For example, monitoring the error rate of a set of books is a good and fast way of evaluating the effects of a change in preprocessing on OCR error rates. It is also a good way of evaluating average OCR errors in different languages.

The specific contributions of this paper are as follows: (i) the recursive text alignment scheme (RETAS) which exploits the unique words approach (Section I) (ii) evaluation of OCR accuracy for a collection of books written in a variety of languages including English, French, German and Spanish (Section V) and (iii) a dataset of books with OCR output and groundtruth texts in the above languages which will be made publicly available [2]. An overview of the alignment problem, the complexity of the proposed method and our conclusions are given in Section II, IV and VI respectively.

II. The Alignment Problem

Standard sequence alignment techniques typically require $O(n^2)$ time and/or space (n = length of the sequences). This can be very expensive since lining up books and their Gutenberg versions requires the alignment of long sequences which can differ considerably. For copyright reasons, the volunteers may remove introductions (some as long as 40 pages). In addition, the scanned versions and the Gutenberg versions may have some differences because they are different editions. These can vary from minor differences to substantial differences. For example, the scanned version may have extra footnotes (which can be considerable for humanities texts) while the Gutenberg version will not have any of these footnotes. There are also scanning errors such as duplicate or missing pages. In addition to all of these, the OCR generated text may be severely degraded due to low document image quality.

Rice [7] proposed to align the ground truth text with the OCR output using Ukkonen's edit distance based string alignment algorithm for evaluating OCR accuracy of synthetic document images. Although Ukkonen's algorithm is very efficient for short sequences, it is too expensive for long sequences especially if there are potentially large gaps such as the books we have. On the other hand Feng and Manmatha's approach [5] and our approach break up the problem in to a number of small quadratic problems each of which can be solved much more efficiently. We discuss this further in the complexity section.

III. The Recursive Text Alignment Scheme (RETAS)

The first stage involves recursively dividing up each input text into smaller pieces and is referred to as the Recursive Stage (Section III-A). In the second stage these short texts are aligned at the word and character level using an edit distance based algorithm to produce a final alignment (Section III-B).

A. Recursive Stage

At each step of the recursion, each segment is divided into smaller ones using a set of unique words called "anchors". Feng and Manmatha [5] use a hash table to find common unique words between aligned segments and then use them as anchors. However, this can introduce errors [3] and hence they added a verification step which involved checking whether the n-grams around the unique words also matched. Here we adopt a different approach. The idea is to find the greatest number of unique words which has the same order in both texts. The problem turns out to be a search for the longest common subsequence between two lists of unique words. It should be noted that in this way we ensure that the resulting segments have the same order in both texts which eliminates the need for a verification step. In order to make the LCS computation more efficient, the unique word sets of the two segments are intersected and only the ones that appear in both texts are used. Unique words in the LCS are used as anchors to split each segment into smaller ones and the recursive step replied to each of these. At the end of the recursion, a large number of short text segments are generated for the final alignment which is performed on the word and then character level.

Figure 1 depicts the proposed alignment scheme for two sample texts. In Figure 1a a small portion of the OCR generated text and its ground truth is shown. Unique words are colored for both texts. Aligning the unique words allows us to determine that the underlined unique words (i.e.,"aliens", "light", "barely") match with each other and are used as anchor points to segment the texts. Thus the text between "aliens" and "light" forms a segment and the text between "light" and "barely" another. Notice that OCR errors generate a number of unique words such as "Plamet"

[2]Data is available at: http://ciir.cs.umass.edu/downloads/ocr-evaluation or http://books.cs.umass.edu/downloads/ocr-evaluation

[3]This is because the texts may have some additional text if they are different editions

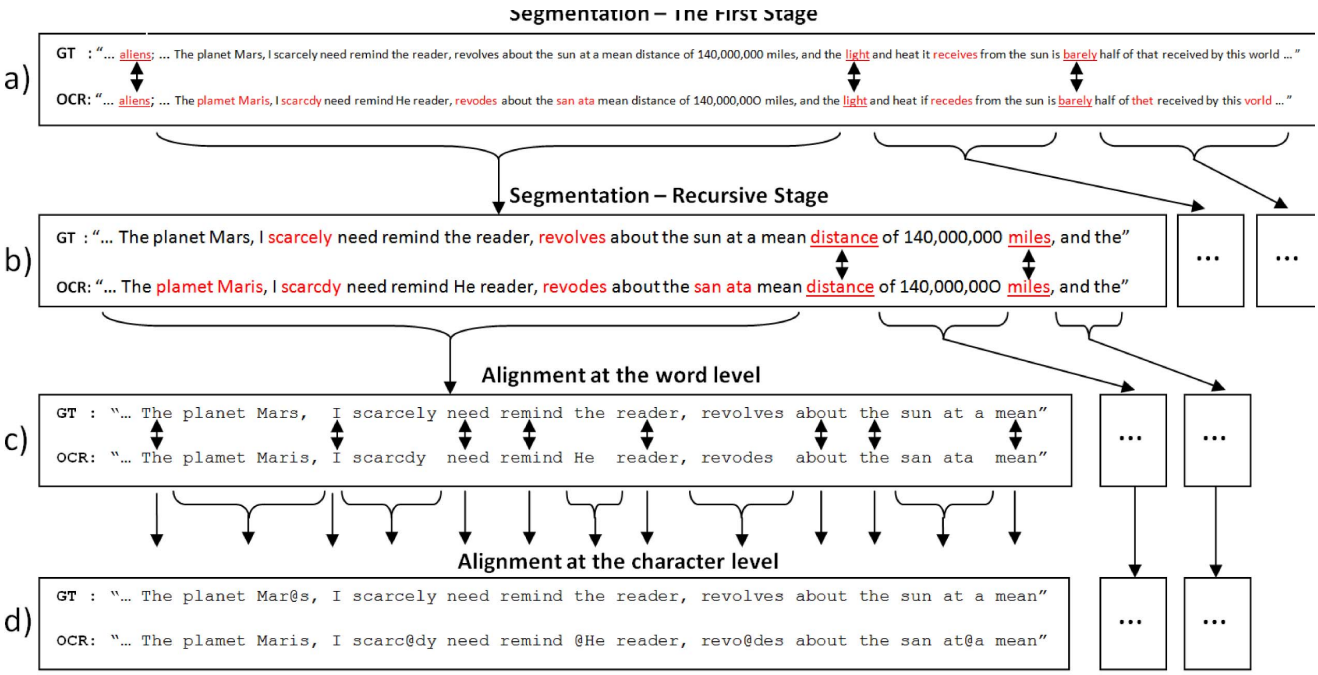

Figure 1. The recursive text alignment scheme (RETAS) depicted for two short texts. "..." stands for skipped content for illustration purposes. Double headed arrows indicate matching words. "@" is a "null" indicator used for signifying character insertion and deletions.

and "thet" but this does not affect the algorithm since they do not align with the other sequence and hence are not used as anchors. Next each segment pair recursively is aligned recursively. Figure 1b depicts the recursion for the text segment between the words "aliens" and "light". Notice that for this segment the words "distance" and "miles" are now unique and can be used for segmenting the text further into shorter segments.

One important point is the stopping criteria for the recursion. One could give a predetermined limit on the depth of recursion or the maximum size for a text segment. In our case, we continue text segmentation recursively until each segment become smaller than a given size K (in our case 200 words). Yet another stopping condition is the time when there does not exist any common unique word for segmentation. One should avoid using a small K since stopwords become unique at the sentence level and this may yield segmentation errors. For example, 'in' and 'on', or 'at' and 'it' may be confused by the OCR engine. At the end of the recursion, a large number of short text segments are generated for the word and character alignment.

B. Word and Character Level Alignment

Corresponding pairs of the short segments produced in the previous stage are now aligned at the character level using an edit distance based algorithm. First, each pair of segments is aligned at the word (Figure 1c) level. The word alignment maps words which are the same across the pair of segments. However, due to OCR errors some words do not align and these are later aligned at the character level to produce the final alignment (Figure 1d). Notice in this case the word "ata" in the OCR string is aligned with "at a" in the ground truth by introducing a null character "@" to correspond to the missing space character. It is observed that aligning words and then characters is more efficient than aligning segments directly at the character level.

At this stage the sequences are short and so the choice of the alignment algorithm usually does not make a significant difference in terms of processing time. In this work we use the standard dynamic programming algorithm for Edit-Distance. The costs for insertion, deletion and replacement are taken as $[1, 1, 2]$ respectively [7].

The length of text segments to be aligned may still be very large after the recursive stage. For example, if the OCR output has large chunks of missing/extra texts or if there are a large number of OCR errors. The former case may occur because of missing text (eg an extra introduction in the scanned version) or because the OCR failed on an entire page because it was blurry. The latter case consists of situations where the OCR error rate is very high. A quick calculation shows that even for fairly high OCR error rates the number of unique words is quite large. For example, for a 200 page book with 500 words/page there are typically between 10 and 15 unique words per page. Even if there is a 50% word error rate the number of unique words is still between 5 and 7 per page (assuming a uniform distribution

of noise) thus implying that most segments will be small. To make sure that the the size of the dynamic programming table in memory is sufficient it is set to have a threshold of (2 million) at both word and character level. This implies that entries up to 2 pages can be aligned even if there is no common unique word. Large stretches without common unique words are more likely due to missing or extra text and hence if the size of the table is likely to be over this limit then the characters are aligned with "null" indicators.

IV. Complexity Analysis

Rice [7] used Ukkonen's algorithm for alignment. This algorithm requires $O(nd)$ time and $O(nd)$ space where n is the length of the sequence in characters and d the edit distance between the two strings. If we take a book which has 200 pages and the scanned version has an extra 20 page long introduction or other matter (not unusual) then d can be of the order of n and hence the complexity in both space and time is $O(n^2)$. In fact while Rice's implementation works very well for a few pages and when there are a small number of changes in the sequences, it fails when there are significant changes between the two sequences as is common in our case. This is not surprising since he designed his algorithm to align short texts to evaluate the OCR accuracy on the page level.

The overall cost of RETAS is characterized by the total cost for the word and character level alignment at the leaf level of the recursion. For the average case, assume that each text segment is divided into k subsegments at each level of the recursion and the length of the text segments at the leaf level is K. Then, the total cost becomes $O(nK)$ since there are n/K text segments each of which takes $O(K^2)$ time to align. For a typical book of about 100,000 words, our alignment is more than two orders of magnitude faster than aligning the whole book as is done by Rice [7]. Our algorithm is also substantially faster than Feng and Manmatha's algorithm [5]. For example, for 200 runs over English books of size 600K characters on average our algorithm took 220 seconds while theirs took 356 seconds. For short books the difference is less obvious.

In theory, the worst case running time is achieved when there are no common unique word between OCR output and the ground truth and in this case the texts have to be aligned using using an exact alignment algorithm at the leaf level (i.e., $K = n$). As mentioned in the previous section this is a very unlikely scenario and never happens in practice.

V. Experiments

The first part of our experiments uses a noise model to create synthetic texts with varying amounts of noise. These texts are used to evaluate the effectiveness of the alignment algorithm. In the second part, we evaluate OCR accuracy of books using the proposed alignment scheme.

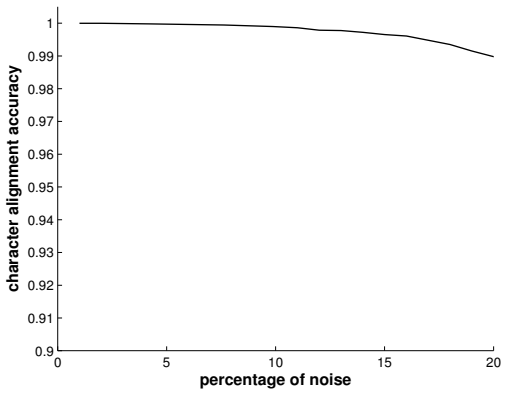

Figure 2. Accuracy of the alignment output versus document noise.

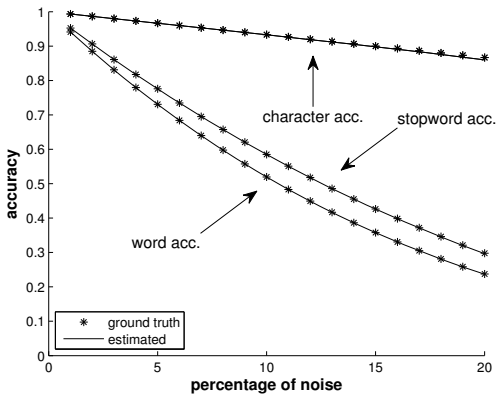

Figure 3. Estimated and the ground truth OCR accuracies for characters, words and stopwords.

A. Verification of the Alignment Scheme

We first look at the behavior of the algorithm when synthetic noise is added to a text. We adopt the noise model introduced in [5]. In a nut-shell, this model applies basic string edit operations on the character level iteratively until the desired amount of noise is reached. For the synthetic experiments, an electronic copy of the book "The Critique of Practical Reason by Immanuel Kant" (English) was obtained from the Project Gutenberg website [6]. The book is converted into a sequence of words each of which is separated by a single space character, letter cases are preserved and all punctuation letters are removed. In this form, the book includes around 350K characters (including spaces) and 63K words. The noise level of a synthetic text is defined by the percentage of randomly inserted, deleted and replaced characters where the distribution of insertion, deletion and replacement operations is $[1/3, 1/3, 1/3]$ for each text. The percentage of changes (noise) is varied from 1 to 20 in steps of 1. The experiment is repeated 100 times with different random seeds and the statistics are averaged.

Table I
ESTIMATED CHARACTER AND WORD OCR ACCURACIES FOR BOOKS IN ENGLISH, FRENCH, GERMAN AND SPANISH FROM THE INTERNET ARCHIVES. PUNCTUATIONS ARE IGNORED.

Dataset	#books	average word length	OCR word accuracy	OCR character accuracy
English	100	4.45	0.934	0.973
French	20	4.91	0.883	0.961
German	20	5.66	0.878	0.949
Spanish	20	4.83	0.900	0.959

Figure 2 shows that the character alignment accuracy is $\geq 99\%$ correct even if there exists 20% noise in the synthetic text. Notice that, for 20% noise, the word error rate is over 75%. The OCR accuracy on real books is actually much higher. Alignment errors occur when an OCR error transforms a unique word to a legal unique word which is also present in the ground truth. For example, transforming "ball" to "call" and could possibly lead to segmentation errors. This is rare since most OCR errors lead to words which are not present in the ground truth.

Figure 3 shows both the ground truth and estimated character, word and stopword accuracies using RETAS. The stopword list consists of 100 most frequent words in English trained using 15 books from the Project Gutenberg. Note that accuracy estimations are almost overlaid over the ground truth values implying that the proposed methodology is successful in estimating OCR accuracies.

B. Evaluation of OCR Accuracy on Real Scanned Books

A number of scanned books in different languages are downloaded from [9] and their OCR accuracy is evaluated. According to the metadata, these books are recognized using ABBYY FineReader 8.0. The ground truth texts are obtained from [6]. For each book, word and character recognition accuracies are estimated. The OCR accuracy metric is defined as follows:

$$OCR_{acc} = \frac{m}{c} \quad (1)$$

where m is the total number of matching characters/words in the alignment and c is the total number of characters/words in the ground truth. This metric accounts for the containment of the ground truth text in the OCR output. The rationale behind this approach is to obtain a statistical evaluation of the OCR accuracy for the portion of the text for which we have ground truth. Note that the scanned text may have extra portions (e.g. an extra introduction) and the metric is not sensitive to such text. With the reasonable assumption that the rest of the book is similar we can assume that the estimated OCR accuracy is true for portions for which we have no groundtruth.

Estimated character and word accuracies are shown in Table I. for four languages using the Latin alphabet. Both word accuracies and character accuracies are directly estimated. English is the most accurately recognized. The word accuracy for Spanish is slightly higher than for French but the character accuracies are reversed (this reflects the fact that word and character statistics depend on language). It is clear that the average OCR word error rate is about 7% for English and more than 10% for other languages. Clearly there is scope for substantial improvement in preprocessing and the OCR itself for the non-English languages. The character accuracy rates even for English do not reach 99% indicating that there is potential for improvement there too.

VI. CONCLUSION

The recursive text alignment scheme (RETAS) is proposed for evaluating the OCR accuracy of books. The basic idea is to scale the whole string alignment problem into manageable size problems. The proposed approach is shown to be effective and fast in this respect. This approach is used to evaluate OCR accuracy of real scanned books in English, French, German and Spanish. Future work includes (i) evaluating OCR accuracy for other languages and scripts, (ii) automatic identification of scanning errors using text alignment.

ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

REFERENCES

[1] I. Z. Yalniz, I. S. Altingovde, U. Güdükbay, and Ö. Ulusoy, "Ottoman Archives Explorer: A retrieval system for digital Ottoman archives," *ACM JOCCH*, vol. 2, no. 3, pp. 1–12, 2009.

[2] J. D. Hobby, "Matching document images with ground truth," *Int. J. on Doc. Anal. and Recog. (IJDAR)*, vol. 1, no. 1, pp. 52–61, 1998.

[3] Y. Xu and G. Nagy, "Prototype extraction and adaptive ocr," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, pp. 1280–1296, December 1999.

[4] T. Kanungo, R. M. Haralick, W. Stuezle, H. S. Baird, and D. Madigan, "A statistical, nonparametric methodology for document degradation model validation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 1209–1223, November 2000.

[5] S. Feng and R. Manmatha, "A hierarchical, HMM-based automatic evaluation of OCR accuracy for a digital library of books," in *JCDL*, 2006, pp. 109–118.

[6] "Project Gutenberg," http://www.gutenberg.org, 2011.

[7] S. V. Rice, "Measuring the accuracy of page-reading systems," in *PH.D. DISSERTATION, UNLV, LAS VEGAS*, 1996.

[8] F. Boschetti, M. Romanello, A. Babeu, D. Bamman, and G. Crane, "Improving ocr accuracy for classical critical editions," in *ECDL'09*, 2009, pp. 156–167.

[9] "The Internet Archive," http://www.archive.org, 2011.