Creating an Improved Version Using Noisy OCR from Multiple Editions

David Wemhoener, Ismet Zeki Yalniz, R. Manmatha
Dept. of Computer Science, University of Massachusetts
Amherst, MA, USA, 01003
{dwemhoen@student, zeki@cs, manmatha@cs}.umass.edu

***Abstract*—This paper evaluates an automated scheme for aligning and combining optical character recognition (OCR) output from three scans of a book to generate a composite version with fewer OCR errors. While there has been some previous work on aligning multiple OCR versions of the same scan, the scheme introduced in this paper does not require that scans be from the same copy of the book, or even the same edition. The three OCR outputs are combined using an algorithm which builds upon an technique which aligns two sequences at a time. In the algorithm a multiple sequence alignment of the scans is generated by stitching together pairwise alignments and is used in turn to construct a corrected text. The algorithm is able to remove OCR errors so long as the same error does not occur in multiple scans. The alignment works even if one of the editions includes an extra long introduction or additional footnotes. This scheme is used to generate improved versions from OCR texts taken from the Internet Archive. The accuracy of the original scans and the composite text are evaluated by comparing them to the version available from Project Gutenberg.**

***Keywords*-OCR error correction; sequence alignment; scanned book collections**

I. Introduction

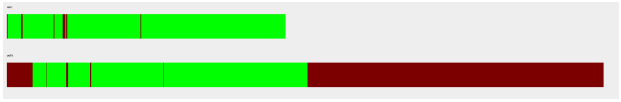
There are currently millions of scanned books available from libraries, universities, and other organizations. These books are available in the form of both page images and optical character recognition (OCR) output. The OCR output is frequently noisy, with errors that can be as small as the alteration of a single letter or as large as an entire page of erroneous characters. OCR correction mechanisms have previously been proposed to correct mistakes in grammar or spelling [1], [2]. A logical strategy for correcting these OCR errors is to take advantage of the large number of books available online that are simply different editions of the same source text or different scans of the same edition. For example, there may easily be tens or even hundreds of versions of a Shakespeare play such as Macbeth or a book such as Jane Austen's Sense and Sensibility. These versions differ in their introductions, footnotes, notes, pagination and formatting but often the main text of the book is the same. Since the OCR errors are likely to be uncorrelated combining them should help reduce the errors.

Such composite editions are useful in many situations. Many online archives such as the Internet Archive contain OCR outputs of books. Such composite editions would have fewer OCR errors and would be useful for improving search and also the reader experience. Humanities scholars often want to look at how multiple versions of a book differ. Since the composite version is produced by aligning with individual versions they can easily look at such differences between versions.

Although each time a text is generated from a scan OCR errors will be introduced, those errors will vary. By using multiple versions of the same text, the scheme is able to take advantage of this variation and remove OCR errors so long as the same error does not occur in multiple scans. The only assumption that needs to be met for correction to be effective is that the texts being aligned all share the majority of the common source text. The process starts by taking the OCR output from a book and viewing it as a sequence of characters in reading order. This paper demonstrates that it is feasible to align the sequences of OCR outputs from multiple different scans of a book to generate a composite version of the main text with fewer OCR errors. While there has been some previous work on aligning three different OCR outputs generated from the same scan [3], the scheme introduced in this paper can generate a multiple sequence alignment of OCR ouputs even if the scans are from different copies of the book, or even if different editions are used. Since each edition is ultimately based on some original version of the book the task is one of recreating this source text by removing unique aspects of each OCR text.

Figure 1 illustrates the challenges that arise because of the differences between three different copies of Wuthering Heights. Copy A is missing ten chapters of the book, Copy B is more or less Wuthering Heights and copy C has an extra book (Agnes Gray) attached to the end of Wuthering Heights. a) shows the overlap between copies A and C and it is clear that the Agnes Gray is extra. C also has an extra introduction at the beginning. b) shows the overlap between copies C and B. Although this looks similar to the previous figure there is a larger overlap between C and B than between A and C since B is a full copy of Wuthering Heights. The thin red (black) lines occur because of approximations in producing the figures or due to actual extra content such as footnotes. All figures are produced automatically using a pairwise alignment algorithm. Later on these texts are aligned to produce a cleaner version of Wuthering Heights.

Sequence alignment is usually done by using a dynamic programming algorithm such as a Longest Common Subsequence (LCS) or a Hidden Markov Model (HMM). For long sequences (such as those produced by books) even pairwise alignments



(a) Pairwise alignment of copy A (top) and C (bottom)

(b) Pairwise alignment of copy C (top) and B (bottom)

Fig. 1. Each figure shows the approximate overlap between a pair of copies of Wuthering Heights. For a given pair, each bar is proportional to the relative length of the book and green (white) indicates overlapping portions while red (black) indicates non-overlapping portions.

of two sequences can be expensive and it is important to do this efficiently. Aligning multiple sequences is even more expensive. We thus focus on a greedy approach to multiple sequence alignment. In this paper we focus only on aligning three texts A, B and C although the approach can be adapted to aligning more sequences. The idea is to take one of the OCR ouputs - say C as pivot - and align each of the other OCR outputs (A and B) separately with it to create two pairwise sequences AC and BC. The common sequence C may then be used to stitch or "zip" all three sequences together to produce a composite output. For this reason the common sequence (C here) will be referred to as a pivot. The zipping process is non-trivial since for example A and B may have characters in common which are missing from the pivot C due to OCR errors. In many situations such errors can be corrected by the zipping process - as will be demonstrated later. The composite output is more accurate than any of the OCR outputs used to construct it.

In order to demonstrate the effectiveness of this scheme, it is applied to two sets of texts (Emily Bronte's Wuthering Heights and Jane Austen's Sense and Sensibility) taken from the Internet Archive. Each set of texts contains three OCR outputs generated from different scans from three versions of the book. The accuracy of the original scans and the composite text are evaluated by comparing them to the version available from Project Gutenberg using the technique proposed in [4]. The Gutenberg versions have been corrected by human editors, making them an effective choice for a ground truth against which to compare other texts to. We demonstrate that on both texts the OCR error rate is improved.

The rest of the paper is organized as follows. Related work is discussed in the next section. Then the multiple sequence alignment and correction framework is described (Section III). This is followed by an experimental section (IV) which shows the results of using this approach with two example sets of texts as a proof-of-concept that the approach works. The conclusion follows.

II. Related Work

OCR error correction is a more challenging compared to the OCR error detection problem since the task is not only to detect OCR errors but also fix them in place automatically. One solution is to use dictionary or n-gram based approaches to first detect OCR errors and then replace them with the most likely word in the dictionary using statistical measures [1]. Although these approaches can reduce the overall OCR error rates for the frequent words of the language, it is likely to corrupt correctly recognized words which are not in the dictionary such as names and places. An alternative approach is to use the context of the text itself to correct misrecognized words. The idea is that OCR errors tend to create words which are not in the vocabulary of the text. One can combine several insights from all of these approaches to help correct OCR errors [2]. However, the success of these approaches are limited if the language models and dictionaries are trained and used on different corpora with different vocabularies such as medical articles and children stories.

If multiple OCR outputs are available for a given document, one can align them to locate and fix OCR errors automatically without using any language specific information [3]. The idea is that the OCR errors are not tightly correlated across different OCR engines although the input document images are the exactly same. The problem is that the multiple sequence alignment is a NP-hard problem. The computational load for an optimal solution exponentially increases as the total number of sequences gets larger. There are several heuristics to make the problem more tractable [5]. The most popular approach is called "progressive alignment" where each pair of sequences are independently aligned first. The alignment outputs are merged one by one starting from the most similar pair [6] to produce the final multiple alignment output. There are also iterative approaches which first creates an initial alignment hypothesis and iteratively improve the alignment by refining it [7]. Yet another approach is to find anchors (or motifs in the context of bioinformatics) to help identify to aligning sections for guiding the alignment efficiently [4], [8]. There are also polynomial time algorithms available to align multiple sequences for the shortest preserving alignment problem which is not applicable in this context [9].

In this particular work, the task is not only to correct OCR errors but also create a composite edition which includes only the conventional (i.e., main body) text for a given set of scanned books. The problem is more challenging than the conventional approach where the document image is assumed to be the same and the output of different OCR engines are aligned. Different editions contain large amounts of additional or missing content which makes the problem complicated.

III. A Multiple Sequence Alignment Framework for Long Noisy Texts

The process of aligning and combining the three OCR outputs can be separated into three stages. The first stage generates pairwise alignments of the three input texts. The second stage builds an alignment of the three texts from the pairwise alignments. The third stage involves taking the multiple sequence alignment and generating a corrected composite text. Figure 1 depicts the steps for converting three OCR

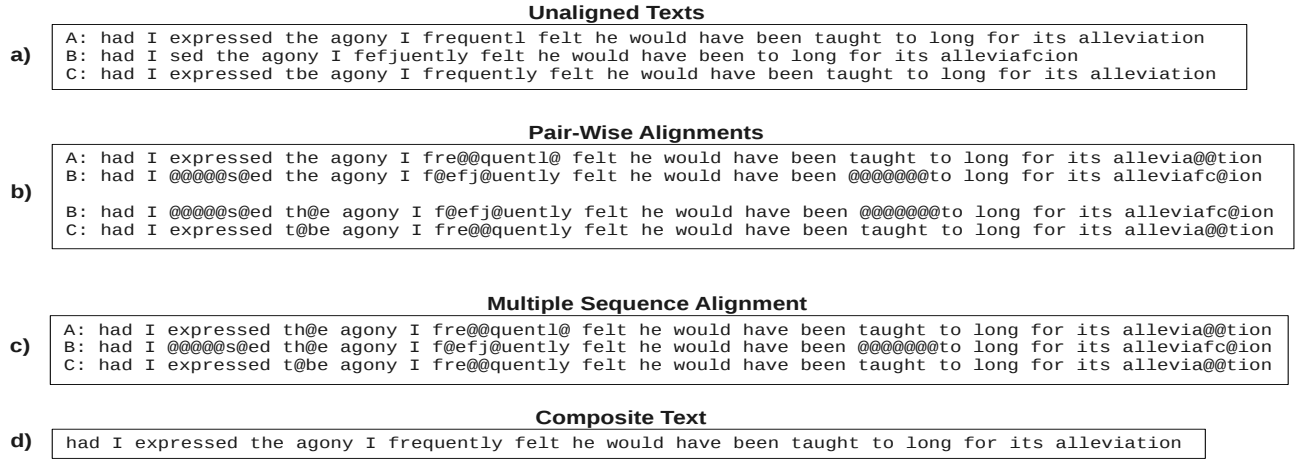


Fig. 2. The alignment scheme demonstrated on a portion of Wuthering Heights. @ represent sections where there was no character that aligned with a character in the other text or texts.

outputs from Sense and Sensibility into a single composite text and is discussed later.

A. Pairwise Alignment

In the first stage of the alignment process one of the three input texts is selected to serve as a "pivot," which will aid in building the multiple sequence alignment later on. This pivot text is separately brought into alignment with the other two texts. It is probably best to choose as pivot the sequence with the lowest OCR error rate. However, this is difficult for two reasons. First, the OCR error rate for a particular scan is not known ahead of time. Second, OCR errors may not be uniformly distributed. Thus, a book which might have a lower error rate in one section as compared to the other two sequences may have a higher error rate in a different section. We, therefore, use an arbitrary sequence as a pivot.

The multiple sequence alignment scheme requires a fast pairwise alignment which can take handle extra or missing text without causing misalignments. Directly running LCS on the two sequences of characters may force align many characters incorrectly. For example if two texts had different pieces of text at the end then the LCS algorithm will tend to incorrectly force align stopwords like "the" in this portion of the text. The recursive text alignment scheme (RETAS) introduced by Yalniz and Manmatha [4] satisfies both these criteria and we modify it for use here. Their sequence alignment scheme is able to quickly bring two texts into alignment by first finding unique words (words which occur once in the book) in each sequence and aligning the unique words using a Longest Common Subsequence (LCS) algorithm. This procedure is then repeated recursively on the sections in between a pair of corresponding aligned unique words. Specifically, words which are unique to each section are now aligned. Finally, when the sections are small enough (400 words) they are aligned using an edit distance based alignment algorithm. This procedure is fast and effective. In addition using the sequence of unique words at the coarse level prevents the alignment from force aligning missing or extra text. The reader is referred to Yalniz and Manmatha [4] for the details.

The texts are first aligned at the word level and then sections still not in alignment are aligned at the character level, allowing for the identification of matching words even when OCR errors have altered characters in one or both words. The following two sentences from Wuthering Heights are aligned at word level with nulls indicating non-matching words. Clearly the words "luxury" and "luxuiy" do not match during word alignment but they differ by only one character.

$$\begin{array}{llll} \text{I let him enjoy the} & \text{luxury} & \text{null} & \text{unannoyed} \\ \text{I let him enjoy the} & \text{null} & \text{luxuiy} & \text{unannoyed} \end{array} \quad (1)$$

The word alignment phase is followed by a character alignment phase which only looks at non-matching words and computes the edit distance between them. When a character in one text fails to align with a character in the other text, a '@' representing a "null" character, is inserted into the other text and the character is aligned with the '@.' This is shown below.

$$\begin{array}{l} \text{I let him enjoy the luxur@y unannoyed} \\ \text{I let him enjoy the luxu@iy unannoyed} \end{array} \quad (2)$$

The alignment now properly reflects that the two words, although they differ from each other by a character, represent the same location in the text. Figure 2b depicts an example of two pairwise alignments, each containing the pivot and one of the other OCR outputs. The pairwise alignment is able to align two books correctly even if they contain missing or extra portions. For example, one book may have footnotes while the other doesn't. It is able to do this because the sequence of unique words ensures that the long range order of the books is preserved and is consistently matched. The alignment also works even if there are large portions of missing or extra text. Failures of alignment occur in certain special circumstances. For example, very high OCR error rates will cause the algorithm to break down. However, such books

are unlikely to be used since large scale errors are hard to correct by any technique.

B. Multiple Sequence Alignment

Once each text has been separately aligned with the pivot, the corresponding sequences in these pairwise alignments are brought into alignment with each other. Since the pivot is common to all the pairwise alignments, the alignments are mapped to each other by matching the characters in the pivot, starting from the first character in both alignments and proceeding forward. The pivot in the two pairwise alignments will be identical except for where null characters have been inserted during the pairwise alignment process. A section of the pivot text may thus appear differently in the two alignments, such as "He was alwa@ys @@@@@qui@ck" and "He@ was always q@uick,". An example of this in a real sequence is shown in Figure 2b where the two versions of the pivot are "the" and"th@e". However, since the pivot is the same original sequence in both cases the only difference is in the number and positioning of null characters. The relationship between the ordering of non-null characters remains constant. Given that all preceding characters have been aligned properly, if the characters in the pivot of both pairwise alignments are equal, they represent the same point in the text. Thus when the "q" is reached in the previous example, it can be assumed to mark the same point in the text without any further examination of the surrounding characters.

Alignment can even succeed when all three texts have different spellings of the same word. The word "frequently" is spelled differently ("frequentl", "fefjuently", "frequently") in each of the three texts in Figure 2b, but in both pairwise alignments it is properly aligned. Note that even if all three versions of the word are incorrect it is possible to correct it if every character of the word is correct in two of the sequences.

There are several scenarios under which the non-pivot texts will contain characters or sequences of characters not present in the pivot text. For sections of the pairwise alignment where there are non-pivot characters which fail to align with the pivot (in Figure 2b these are the sections where the pivot has a value of '@'), the pairwise alignments are brought into alignment by aligning the non-pivot characters. Thus the word "taught" which occurs in the first and third sequences in Figure 1 can successfully be aligned even though the word is entirely absent from the pivot text. Another example is the word "expressed" which is mostly missing from the pivot text but present in the other texts. It may also happen that the non-pivot text contains a character introduced due to OCR error, in which case there will be no corresponding character in the pivot text, as is the case with the misspelled "the" in text C which resulted in the insertion of an '@' into the pivot.

C. Error Correction

In the final phase a composite version of the texts (Figure 2d) is constructed from the multiple sequence alignment (Figure 2c). For each aligned triplet of characters, the character to be inserted into the combined text is chosen by majority vote. Since at least two of the characters in any triplet must be equal (they may be null), there will always be a majority choice. If the chosen character is a null, then nothing is inserted into the text. This only occurs if one of the texts contained a character or sequence of characters unique to that text. Thus the 'b' in 'tbe' from text C does not appear in the composite text, but the 'y' at the end of "frequently" which doesn't occur in text A does appear in the composite text. In fact, it is not necessary for a word to be correct in any of the sequences. However, each column must contain at least two correct characters for the word to be recognized correctly. In this manner, both OCR errors and edition-specific words or sections are excluded from the composite text.

Many books have running (page and chapter) headers and these can also be effectively removed by the error correction process, which is useful since it would be otherwise difficult to remove them from the text. This is possible if each version of the book has a different running header or the running header occurs at a different place in the sequence. Often two copies of the book have different formatting and pagination (see for example Table I) ensuring that page and chapter headers occur at different places. For example, it may be hard to remove all instances of "Wuthering Heights" with some kind of preprocessing tool since these words are present both in the header and in the text. OCR errors also would make alternative techniques to remove running headers more difficult.

TABLE I
CONTENT AND SIZE INFORMATION FOR SEVERAL SCANNED VERSIONS OF SENSE AND SENSIBILITY (SS) AND WUTHERING HEIGHTS (WH).

Book	Edition	Word count	Character count	Page count	Extra/ missing text
SS	1833	130493	717862	368	complete
SS	1864	124459	686867	353	complete
SS	1844	121733	666421	475	complete
WH	1896	96219	539771	327	missing 10 chpts
WH	1900	124126	667128	299	complete
WH	1848	208117	1116489	643	contains "Agnes Grey"

IV. EXPERIMENTS

We apply our approach to multiple sequence alignment to two sets of example books and compare the accuracy of the composite texts we generate to "master" versions of each book.

A. Datasets

To test the algorithm, it was applied to two collections of OCR outputs. The first collection was of three different editions of Emily Bronte's Wuthering Heights. The second collection was of three different editions of Jane Austin's Sense and Sensibility. Table I shows statistics about the scanned books and their editions used in the experiments. These books are downloaded from the Internet Archives website [10]. The Internet Archive also provides the OCR output (based on a commercial OCR).

Only one of the three books from which the OCR outputs for Wuthering Heights were generated includes the entirety

of the story with no additional texts. One book also contains the novel Agnes Grey, and the other only has the first 25 chapters of Wuthering Heights. All of the copies have word-level accuracies of less than 92% and one of the copies has a word-level accuracy of less than 82%.

B. Implementation Details

Before the texts are aligned they are preprocessed. The preprocessor removes punctuation (“.,;:=-/’‘&|$#@!%^*}{()[]_"\<>?~+”) as punctuation marks are frequently incorrectly recognized/inserted during the recognition process and would interfere with alignment. Numerical letters are also removed since they often correspond to page numbers which is not consistent across books. Those page numbers are quite likely to be unique in the context which may mislead the recursive text alignment scheme. In the case of hyphenation due to a line break, the text preprocessor will connect the two words. Thus ”cer-” and ”tainly” becomes ”certainly.” Since the location of line breaks varies between versions of the book, reconnecting words broken between lines aids in comparing the texts. In this particular application, the case is folded, thus terms such as ”Cat” and ”cat”, which may be the same word processed differently depending on the quality of the scan used, would be treated as the same word.

C. Evaluation

The character accuracies are estimated by pairwise aligning the noisy texts with their Project Gutenberg versions (contains error-free e-books containing only the main text of the books [11]) at the word and character levels as described in [4]. The character level accuracy is determined by the total number of matching characters in the alignment divided by the total number of characters in the ground truth text. In Table II, it is seen that the OCR accuracy of the composite texts have a greater word accuracy (about 4%) than the book with the highest OCR accuracy among all the editions. Although one of the copies of Wuthering Heights contained a large amount of extraneous text and another was missing a significant portion of the text (as shown in Figure 1, the composite text is more accurate than any of the editions and it includes the complete copy of the original work. The most accurate OCR output of Wuthering Heights had an accuracy of 88.47%, while the combined text has an accuracy of 92.40%, which is an increase of 3.93 percentage points. For Sense and Sensibility, the composite text had an accuracy of 95.39%, which is 4.14 percentage points greater than the accuracy of 91.25% for the most accurate edition. It is interesting to note that even with high character accuracies one an have low word accuracies. For example, several of the books have character accuracies in the mid 90% range but the word accuracies are much lower. If OCR errors are spread out versus occuring in runs then word accuracies are likely to be lower. For example, consider a document with 10 words each of which has 5 characters each and assume there are 5 character errors. Then the character accuracy is 90%. Depending on how the errors are distributed the word accuracy can vary from 90% to 50%. On a single core using a desktop computer with 3.4GHz processor, it took 8.79 and 16.12 seconds respectively for generating the composite editions for Sense and Sensibility and Wuthering Heights.

TABLE II
ESTIMATED CHARACTER AND WORD OCR ACCURACIES FOR SCANNED AND CORRECTED VERSIONS OF SENSE AND SENSIBILITY AND WUTHERING HEIGHTS.

Book	Edition	OCR word accuracy	OCR character accuracy
Sense and Sensibility	1833	0.8130	0.9368
Sense and Sensibility	1864	0.9125	0.9760
Sense and Sensibility	1844	0.9111	0.9541
Sense and Sensibility	Composite	0.9539	0.9885
Wuthering Heights	1896	0.7346	0.7482
Wuthering Heights	1900	0.8343	0.9428
Wuthering Heights	1848	0.8847	0.9713
Wuthering Heights	Composite	0.9240	0.9765

V. CONCLUSION

The concept of generating an error-corrected composite version from multiple editions has been demonstrated for sample scanned books. The proposed approach uses a fast text alignment scheme to align pairs of texts at the first step. Multiple sequence alignment is generated by combining the output of pairwise alignment and a voting scheme is used to correct OCR errors. It is shown that the composite texts have significantly higher OCR accuracies compared to the other editions without any additional or redundant text in the form of introduction, publisher details, vocabulary etc. Future work includes (i) combining larger number of editions for creating cleaner composite texts, (ii) mapping extra or missing portions of texts across editions, and, (iii) improving the OCR accuracies further using other sources of contextual or linguistic information such as grammar.

REFERENCES

[1] K. Kukich, “Technique for automatically correcting words in text,” *ACM Computing Surveys*, vol. 24, no. 4, pp. 377–439, Dec. 1992.

[2] X. Tong and D. A. Evans, “A statistical approach to automatic OCR error correction in context,” in *Fourth Workshop on Very Large Corpora (WVLC-96)*, 1996, pp. 88–100.

[3] 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.

[4] I. Z. Yalniz and R. Manmatha, “A fast alignment scheme for automatic ocr evaluation of books,” in *ICDAR*, 2011.

[5] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and Nucleic acids*. Cambridge University Press, 1999.

[6] J. D. Thompson, D. G. Higgins, and T. J. Gibson, “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice,” *Nucleic Acids Research*, 1994.

[7] M. Hirosawa, Y. Totoki, M. Hoshida, and M. Ishikawa, “Comprehensive study on iterative algorithms of multiple sequence alignment,” *Computer Applications in the Biosciences*, vol. 11, no. 1, pp. 13–18, 1995.

[8] R. F. J. P. O. S. S. A.L. Delcher, S. Kasif, “Alignment of whole genomes,” *Nucleic Acids Research*, vol. 27, p. 23692376, 1999.

[9] S.-H. Sze, Y. Lu, and Q. Yang, “A polynomial time solvable formulation of multiple sequence alignment,” in *RECOMB*, 2005, pp. 204–216.

[10] “The Internet Archive: digital library,” http://www.archive.org, 2013.

[11] “The project Gutenberg: free e-books,” http://www.gutenberg.org, 2013.