Evaluation of OCR Text Correction by Crowdsourcing on a Historic Newspaper Archive

Megha Gupta and Haimonti Dutta*
Department of Computer Science, IIIT-Delhi
*Affiliated to The Center for Computational Learning Systems, Columbia University, New York
(megha1124, haimonti)@iiitd.ac.in

***Abstract*—Optical Character Recognition (OCR) is a common method of digitizing printed texts so that they can be electronically searched, stored compactly, displayed on-line, and used in text mining applications.**
The text generated from OCR devices, however, is often garbled due to variations in quality of the input paper, size and style of the font and column layout. This adversely affects retrieval effectiveness and techniques for cleaning the OCR need to be improvised. Often such techniques involve laborious and time consuming manual processing of data.
This project deals with a subset of historical newspaper articles in the holdings of the California Digital Newspaper Collection [?] which have been OCR-ed. Patrons of the [?] read the Amador Ledger on a regular basis and correct OCR errors as they come across them.
This work can be used to determine whether the OCR text is clean enough to be used for thorough text search. The creation and analysis of this corpus will enable advanced search mechanisms on these holdings making them more useful to the general public.

***Index Terms*—OCR; newspapers; historic; crowdsourcing;**

I. Introduction

The electronic conversion of scanned images of handwritten or printed text into a machine encoded text is widely done using Optical Character Recognition devices. It finds most successful applications in the field of machine Learning, Artificial Intelligence and Pattern recognition.
OCR deals with the problem of recognising optically generated characters be it offline or online. Its performance directly depends on the quality of input document. The more constrained the input is the better will be the performance of the system. But when it comes to unconstrained handwriting, the performance is far from satisfactory.
The main application areas for [**?**] like Automatic number plate readers, form readers, Signature verification and identification fall into three categories that is Data Entry, Text Entry, process automation.
This project deals with printed text in the form of Historical Newspaper Articles in the holdings of [**?**]. One such newspaper, The Amador Ledger published in the early 1900s by the Amador Publishing Company appealed to the community's interests by covering issues unique to gold mining.
Patrons of the [**?**] continue to be interested in studying about the status of the local mining industry and consequently read the Amador Ledger on a regular basis even to this day and correct [**?**] errors as they come across them.
The corrections made by the patrons generate logfiles that maintain the history of what users have edited which further helps in generating the Corrected OCR corpus for evaluation and analysis.
Experiments are performed on both the datasets using the same Query set. The ranked documents retrieved from both the sets are compared using a statistical measure called Spearman's Rank Order Correlation. It measures the strength of association between two ranked variables.
This work can be used to determine whether the OCR text is sufficient enough to be used for the thorough text search and analysis and does it meet the user expectation without actually enhancing and enriching the text.

A. Project Participants

This is an individual project.

B. Datasets and Software Used

1) Logfiles: Logfiles were generated using a third party software for user text correction issued by Digital Libraries Consulting [**?**]. Using this software, patrons could edit the incorrect text, producing a logfile in xml format containing tags like
<OldTextValue>largest Plant in the World.</OldTextValue>
<NewTextValue>Largest Plant in the World.</NewTextValue>

The names of the logfiles follow a convention as the first two letters represent the initials of the newspaper followed by the date as yyyymmdd.
For example AL19000105-changes.log includes Amador Ledger, 1900-01-05. There were in total 234 logfiles.

To get an idea of the number of corrections made by user per logfile, we created a histogram which is shown below.

The errors rectified by the users can be categorized as Spellcheck Error, Capitalization Error, Punctuation Error, Addition of new words, Omission of Garbled Text, etc.

```
:TextCorrectedBlock pageOID="AL19000105.1.1" blockID="P5_TB00040" userID="[redacted]">
  <TextCorrectedLine lineID="P5_TL00234">
    <OldTextValue>largest Plant in the World.</OldTextValue>
    <NewTextValue>Largest Plant in the World.</NewTextValue>
  </TextCorrectedLine>
</TextCorrectedBlock>
:TextCorrectedBlock pageOID="AL19000105.1.1" blockID="P5_TB00045" userID="[redacted]">
  <TextCorrectedLine lineID="P5_TL00272">
    <OldTextValue>stem of tho plant is as strong as an or-</OldTextValue>
    <NewTextValue>stem of the plant is as strong as an or-</NewTextValue>
  </TextCorrectedLine>
  <TextCorrectedLine lineID="P5_TL00275">
    <OldTextValue>habitants of tho South Sea Islands,</OldTextValue>
    <NewTextValue>habitants of the South Sea Islands,</NewTextValue>
  </TextCorrectedLine>
```

Fig. 1. Log File

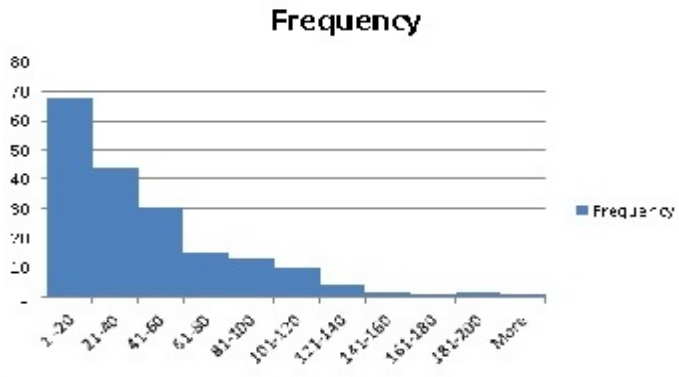

Fig. 2. Histogram

2) Raw OCR corpus: This corpus had 190 files with the original OCR-ed text. It contains errors due to inaccurate conversion from printed to digitized text by OCR devices. The files in this corpus were extracted from the the website [**?**] using Python scripts. The general idea was to decode the names of the logfiles to get the name and date of the newspaper. Further these name and dates were formulated into a URL. The url for a particular issues has a pattern that changes only with the date of the issue. The issue we dealt with in our corpus was Amador Ledger, ranging from 1900-01-05 to 1910-12-30. For instance, AL19000105-changes.log was converted to Amador Ledger, 1900-01-05 which was further translated to http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-1/ocr.txt

3) Corrected OCR Corpus: This corpus incorporated users corrections that were recorded in logfiles. It was generated by replacing the old text in the Raw OCR corpus with the new text given in the logfiles. Python scripts were used to integrate both the datasets, that are the logfiles and the Raw corpus to create a new Corrected corpus containing 190 files.

4) Query Set: This dataset was created by randomly picking words only from the Corrected corpus as this corpus is an enhanced copy of Raw Corpus. So the keywords include

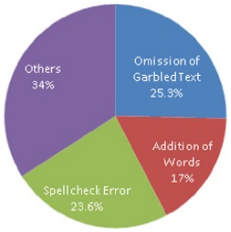

Fig. 3. Error Classification

ocr-ed text that was not touched by the patrons and the text that was corrected by them through crowd sourcing.

5) Software: PyLucene 3.6.2, a Python extension for accessing Java Apache Lucene was used as an IR software library for enabling full text indexing and searching capabilites. Inverted Index was built on each corpus with the fields stored as file name, file path and file contents making retrieval faster.

II. Problem Statement

This research is primarily focussed on retrieval effectiveness of OCR data versus Corrected data. We are trying to figure out what are the effects of noisy data on text retrieval. To know the difference between the results we need to find a heuristic measure that determines the difference between the retrieval effictiveness of OCR data versus Corrected Data. The aim of the project is to create and analyse the corrected OCR corpus using a metric that would measure the difference between the retrieval effectiveness of both the corpora, that is Raw OCR versus Corrected OCR.

III. Methodology

A. Preprocessing & Data Generation

1) Logfiles: Downloaded the Logfiles from the file server using pscp command. These Logfiles (xml format) maintain the history of old OCR text between the tag <OldTextValue></OldTextValue>
and corresponding corrected text between <NewTextValue></NewTextValue> tag.

2) Raw OCR Corpus: These logfile names were decoded into Newspaper name and Date of Newspaper.
For example, Logfile name, AL19000105-changes.log was converted to Amador Ledger, 1900-01-05
Then each of the decoded name was translated into eight individual URLs.
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-1/ocr.txt
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-2/ocr.txt
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-3/ocr.txt
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-4/ocr.txt

http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-5/ocr.txt
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-6/ocr.txt
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-7/ocr.txt
http://chroniclingamerica.loc.gov/lccn/sn93052980/1900-01-05/ed-1/seq-8/ocr.txt
where seq-1 represents the sequence of the pages of the newspaper. Finally, we downloaded the Raw OCR text from each of these URLs from [**?**] with the help of Python scripts to build the Raw OCR corpus.

3) Corrected OCR Corpus: Replacing the incorrect text with the corrected text from the logfiles in the Raw OCR Corpus, we obtain the Corrected OCR Corpus. The old and its corresponding new text were stored as key value pairs of a dictionary. Once all the pairs were generated from a log file, the text from the Raw Corpus matching with keys of the dictionary was replaced with the values corresponding to the matched key.

B. Building IR Model

PyLucene, is a wrapper around Java Lucene that allows full functionality of lucene in Python. Its goal is to allow the use of Lucene's text indexing and searching capabilities from Python.
We have used PyLucene 3.6.2 in the project for Indexing and retrieving documents. Firstly, inverted indexes were created for both the datasets. An Inverted Index is an inside out arrangement of documents where terms take the center stage and each term points to a list of documents that contain it. Index file contain fields, that includes, file name, file path and content.
Lucene's logical View of Index files features segments which contain indexed documents. Segments can be searched independently and the number of segments are determined by the number of documents to be indexed and the maximum number of documents in each segment.

C. Query Set generation

The Query set was formulated by randomly picking keywords from each of the Corrected OCR document. Each Query comprise of a single word keyword, for example "bapism", "eparmen", "elecors", etc. The number of keywords in the query set is around 200.

D. Experimentation

The experiments were run on the indexes created on both the datasets using the same query set. Once the query set was generated, it was passed through the searching script that retrieves the documents containing the keywords.
The documents retrieved are ranked according to the frequency of the keyword present in the document. Higher the frequency, higher the ranking order.

E. Results

The ranked retrieved results (documents) from both the corpus for a query "January" are shown below :
From Corrected OCR

Top 10 matching Documents
name:1900-01-12.txt
name:1902-01-17.txt
name:1900-01-05.txt
name:1904-01-08.txt
name:1905-01-27.txt
name:1902-02-07.txt
name:1904-01-29.txt
name:1904-12-30.txt
name:1905-01-13.txt
name:1908-08-14.txt

From Raw OCR

Top 5 matching Documents
name:1900-01-12.txt
name:1902-01-17.txt
name:1900-01-05.txt
name:1904-01-08.txt
name:1905-01-13.txt

F. Evaluation

The metric we used to compare the ranked retrieved documents is Spearman's rank correlation coefficient(P). It is a nonparametric measure of statistical dependence between two variables. It assesses how well the relationship between two variables can be described using a monotonic function. If there are no repeated data values, a perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other.
Spearman's coefficient, like any correlation calculation, is appropriate for both continuous and discrete variables, including ordinal variables.
In applications where duplicate values are not present, the spearman's coefficient can be calculated using the below formula.
The ranking order of the documents retrieved from the Corrected OCR Corpus (Y_i) is shown in Rank y_i whereas the order for the Raw OCR (X_i) is given as Rank x_i . When the retrieved documents from both the dataset differ, then ranking of the Corrected OCR will be treated as a benchmark and Raw OCR will be ranked according to it. But the actual rank of the Raw OCR will be the sorted order of the rank given in accordance with the benchmark.

Differences d_i = x_i - y_i between the ranks of each observation on the two variables are calculated, and P is

given by: P = 1 - $\frac{6\sum d_i^2}{n(n^2-1)}$

Following table shows the evaluation of P for a single query "January" (Best Case).

Y_i	X_i	Rank y_i	Rank x_i	d_i	d_i^2
12-01-1900	12-01-1900	1	1	0	0
17-01-1902	17-01-1902	2	2	0	0
05-01-1900	05-01-1900	3	3	0	0
08-01-1904	08-01-1904	4	4	0	0
27-01-1905	13-01-1905	5	9(5)	0	0

$\sum d_i^2$ = 0
P = 1 - $\frac{6*0}{5*24}$
P = 1

For another Query "Jackson", the evaluation of P (Worst Case) is as follows:

Y_i	X_i	Rank y_i	Rank x_i	d_i	d_i^2
12-01-1900	21-10-1910	1	10(4)	-3	9
09-12-1910	09-12-1910	2	2(1)	0	0
08-06-1900	23-09-1910	3	7(2)	1	1
01-04-1904	02-12-1910	4	8(3)	1	1
25-05-1900	23-12-1910	5	11(5)	0	0

$\sum d_i^2$ = 11
P = 1 - $\frac{6*11}{5*24}$
P = 0.45

The average value of Spearman's ranked correlation cofficient can be determined by averaging the Best and Worst case.
P = $\frac{0.8+0.45}{2}$ = 0.625

IV. Conclusion

The sign of the Spearman correlation indicates the direction of association between X (Raw OCR results) and Y (Corrected OCR results). If Y tends to increase when X increases, the Spearman correlation coefficient is positive. If Y tends to decrease when X increases, the Spearman correlation coefficient is negative. A Spearman correlation of zero indicates that there is no tendency for Y to either increase or decrease when X increases. The Spearman correlation increases in magnitude as X and Y become closer to being perfect monotone functions of each other. When X and Y are perfectly monotonically related, the Spearman correlation coefficient becomes 1.
The average value of Spearman's ranked correlation coefficient calculated by our experiments is 0.625 which can be interpreted as the association between the two corpora is not very strong but the positive value shows that if Raw OCR increases then Corrected OCR will definitely increase.

To use F-measure we need to have the record of relevant documents for each query. Every user has its own notion of relevance as it is a subjective measure.

V. Future Work

Due to lack of time, we could not incorporate the users feedback or judgement regarding relevant and non relevant documents for particular query. We would like to extend our evaluation using F-score.
Also, we would like to enhance our search capabilities from single word query to multiple words or phrases.

Acknowledgment

I would like to express my gratitude to my Advisor, Dr. Haimonti Dutta for her support, patience and encouragement throughout the course of the project.

VI. Bibiliography

References

[1] Chronicling America. http://chroniclingamerica.loc.gov/, June 2009.
[2] California Digital Newspaper Collection. http://cdnc.ucr.edu/cdnc, June 2009.
[3] New Zealand Digital Library Consulting, University of Waikato in Hamilton. http://www.dlconsulting.com/, January 1990.
[4] Line Eikvil. Optical character recognition. 1993.