Proceedings of the 2015 INFORMS Workshop on Data Mining and Analytics
M. G. Baydogan, S. Huang, A. Oztekin, eds.

Evaluation Of Spell Correction On Noisy OCR Data

Aayushee Gupta
Indraprastha Institute of Information Technology
Delhi, India[1]
aayushee1230@iiitd.ac.in

Haimonti Dutta
Assistant Professor, Management Science And Systems
School of Management University
Buffalo, USA
haimonti@buffalo.edu

Abstract

Optical Character Recognition (OCR) of historical text often leads to several kinds of spelling errors. Existing spell correction algorithms don't present a rigorous performance evaluation of the spell correction process. In this paper, we present a novel N-gram based algorithm for checking the accuracy of spell correction which can handle noisy and cleaned text of different lengths. First, an edit distance based spell correction algorithm is presented which forms the basis of the evaluation mechanism. The goal of the Spell Correction Evaluation (SCE) algorithm is to measure accuracy of the text corrected by the spell correction algorithm against manually corrected data. N word grams are used to make word-to-word correspondences between the corrected text and the noisy OCR text - each token is then annotated as a True/False Positive/Negative of the correction process. The algorithm relies on appropriately choosing a window of N-words and aligning them in three parallel corpora - noisy OCR, corrected and manually cleaned text (ground truth). Empirical results of spell correction on 50 news articles from the "The Sun" newspaper, Nov Dec 1894 are presented and the SCE algorithm evaluates its accuracy to be 72.7

Keywords: OCR, Spell Correction,Spell Correction Evaluation,Historical Newspaper Archives

1 Introduction

Introduce noisy text–OCR text, its applications. Eg. handwriting recognition, historical archives. Problems in OCR text. Requirement of spell correction. Limited complete spell correction evaluation strategies. Novelty? Paper organization.

2 Data Description

This section describes the dataset used for spell correction on OCR data. Following sections provide details of data source, characteristics and some data statistics.

2.1 Data Source

The dataset has been taken from Chronicling America. *Chronicling America*[2] is an initiative of the National Endowment for Humanities (NEH) and the Library of Congress (LC) whose goal is to develop an online, searchable database

[1]Currently working as Assistant Professor with the CSE Department at Jaypee Institute of Information Technology, Noida, India.

[2]http://chroniclingamerica.loc.gov/

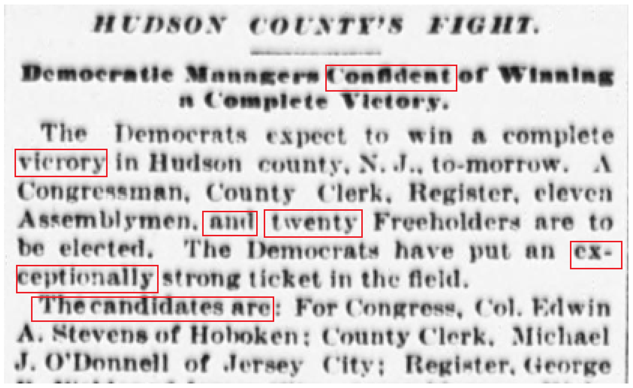
of historically significant newspapers between 1836 and 1922. The New York Public Library (NYPL) is part of this initiative and has scanned 200,000 newspaper pages published between 1890 and 1920 from microfilm.

In order to make a newspaper available for searching on the Internet, the following processes used in [6] must take place: (1) the microfilm copy or paper original is scanned; (2) master and Web image files are generated; (3) metadata is assigned for each page to improve the search capability of the newspaper; (4) OCR software is run over high resolution images to create searchable full text and (5) OCR text, images, and metadata are imported into a digital library software program. The scanned newspaper holdings of the NYPL offers a wealth of data and opinion for researchers and historians.

The newspaper titles and digitized pages available through the Chronicling America website can be searched using the OpenSearch protocol[3]. Unfortunately, the current search facilities are rudimentary and irrelevant documents are often more highly ranked than relevant ones. The newspapers are scanned on a page-by-page basis and article level segmentation is poor or non-existent; the OCR scanning process is far from perfect and the documents generated from it contains a large amount of garbled text. In a bid to serve its patrons better, the New York Public Library employed human annotators to clean headlines of articles and text, but the process of manually reading all the old newspapers article-by-article and cleaning them soon became very expensive.

2.2 Data Characteristics

An individual OCR text article has at least one or more of the following types of spelling errors:

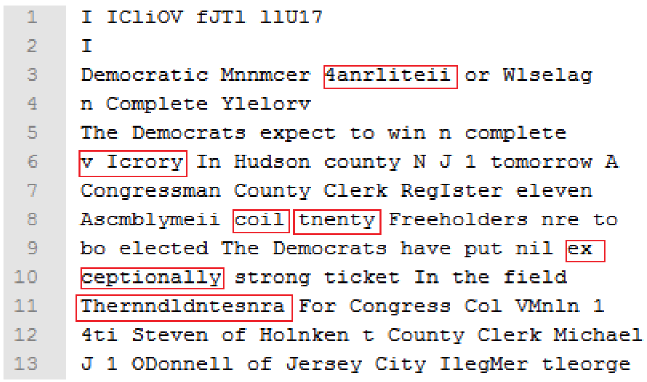


HUDSON COUNTY'S FIGHT.

Democratic Managers Confident of Winning a Complete Victory.

The Democrats expect to win a complete vicrory in Hudson county, N. J., to-morrow. A Congressman, County Clerk, Register, eleven Assemblymen, and twenty Freeholders are to be elected. The Democrats have put an ex-ceptionally strong ticket in the field.

The candidates are: For Congress, Col. Edwin A. Stevens of Hoboken; County Clerk, Michael J. O'Donnell of Jersey City; Register, George



```
I ICliOV fJTl 11U17
I
Democratic Mnnmcer 4anrliteii or Wlselag
n Complete Ylelorv
The Democrats expect to win n complete
v Icrory In Hudson county N J 1 tomorrow A
Congressman County Clerk RegIster eleven
Ascmblymeii coil tnenty Freeholders nre to
bo elected The Democrats have put nil ex
ceptionally strong ticket In the field
Thernndldntesnra For Congress Col VMnln 1
4ti Steven of Holnken t County Clerk Michael
J 1 ODonnell of Jersey City IlegMer tleorge
```

Figure 1: Scanned Image of a Newspaper article (left) and its OCR raw text (right)

- **Real word errors** include words that are spelled correctly in the OCR text but still incorrect when compared to the original newspaper article image. For example: In Figure 1, the word "coil" has been correctly spelled in the OCR text but should have been "and" according to the original newspaper article.
- **Non-real word errors** include words that have been misspelled due to some insertion, deletion, substitution or transposition of characters from a word. For eg. In Figure 1, the word "tnenty" in the OCR text has a substitution error ('n' should have been 'w') which is actually "twenty" according to the original newspaper article.
- **Non-word errors** include words that have been spelled incorrectly and are a combination of alphabets and numerical characters. For example: In Figure 1, the word "4anrliteii" which is a combination of alphabets and number and should have been "confident" as per the original newspaper article.
- **New Line errors** include words that are separated by hyphens where part of a word is written on one text line and remaining part in the next line. For example: In Figure 1, the word "ex-ceptionally" where "ex" occurs on one line while "ceptionally" in the next and due to no punctuation in the text, they are treated as separate words in OCR text.

[3] http://www.opensearch.org/Home

•**Word Split and Join errors** include words that either get split into one of more parts or some words in a sentence get joined to a make a single word. For example: In Figure 1, the word “Thernndldntesnra” in the OCR text is actually a combination of three words “The candidates are” while the words “v Icrory” are actually equivalent to a single word “victory” when compared with the original news article.

2.3 Data Statistics

The OCR text available from Chronicling America website is on a page by page level and no article level segmentation is provided. OCR text dataset is therefore, taken from a PostgreSQL database where article level segmentation of page-level OCR text from Chronicling America is available for two months of articles of “The Sun” newspaper from November-December 1894 consisting of 14020 news articles with a total of 8,403,844 tokens. The newspaper database ER diagram [4] is used to extract the required articles text from the database by dumping complete dataset and extracting individual articles linetext based on their unique ID. The individual text articles generated from the database do not have any punctuation and contain a large amount of garbled text containing above mentioned OCR errors.

3 Spelling Correction

Several kinds of spelling errors exist in the data as described in the previous section. This section first provides a brief review of spelling correction algorithms that exist in literature (Section 3.1); Section 3.2 describes the spelling correction algorithm used in this research and evaluation results on the OCR dataset are presented in Section 3.4 followed by discussion in Section 4.

3.1 Related Work

Kukich [8] comprehensively discusses various spelling correction techniques based on Non word, Isolated word and Real word spelling errors. N-gram analysis, Dictionary lookup and Probabilistic techniques are used for correcting isolated and nonword errors while Context-Dependent techniques are used mostly for correcting real word errors including the correction of word split and join errors [7].

N-gram techniques work by examining each n-gram in the text string and comparing against a pre-compiled table of n-gram statistics to retrieve the correct word while Dictionary look up techniques directly check whether the text string appears in the dictionary using string matching algorithms. Both techniques require a dictionary or a large text corpus and take frequency of n-grams or word occurrence into account in order to find the correct spelling [15], [13]. Probabilistic techniques use transition and word confusion probabilities to estimate likelihood of the correction in order to rank and retrieve correct word spelling.

On the other hand, Context-dependent techniques require contextual information and use either extensive NLP techniques or Statistical Language Modeling (SLM) for spelling correction. Bassil and Alwani [2] use Google 1-5 gram word dataset to gain context information in order to determine the correct words sequence in the text for correction. Tong and Evans [17] use SLM approach involving information from letter n-grams,character confusion and word bi-gram probabilities to perform context sensitive spelling correction obtaining a 60 percent error reduction rate.

All these spelling correction techniques have developed over time and have been used in combination to achieve improved accuracy [3]. [1] use a combination of Google suggestions, LCS and character confusion probabilities for choosing the correct spelling on a small set of historical newspaper data and achieve recall and precision of 51% and 100% respectively.

Edit distance approach, suggested initially by Wagner and Fischer [19], is a dictionary lookup approach commonly used for OCR data correction because of the large number of substitution errors in OCR data [8] [5] which can be corrected using this technique. String edit distance approaches with faster correction are discussed in [10], [14] with variants like Levenshtein automata and normalized edit distance.

[4]https://power.ldeo.columbia.edu/twiki/pub/Incubator/BodhiDBDesign/Final ERD.pdf

Personal name spelling correction has also been studied separately including comparison among various techniques indicating that there is no one single technique that outperforms all others though pattern matching techniques result in better matching quality compared to phonetic encoding techniques [5]. Almost all the personal name spelling correction techniques use personal names directory/dictionary for matching against wrongly spelled names in the dataset or queries in case of People Search [18].

We use edit distance algorithm for spelling correction because of its speed and ability to correct OCR errors compared to the n-gram approach [4]. Context-dependent spelling correction is not used because of unavailability of n-gram words corpus or ground truth dataset containing OCR and true word pairs. Our edit distance algorithm also uses an enhanced dictionary for look up to give significance to personal names spelling correction in the dataset.

3.2 Spelling Correction Algorithm

The Edit Distance algorithm based on Levenshtein distance [9] has been used for spelling correction. It is an isolated word correction technique that uses dictionary based-look up method and distance between strings for matching the text and correcting it. An "edit distance" corresponds to the minimum number of insertions, deletions, and substitutions required to transform one string into another. The algorithm corrects Non-Real Word spelling errors up to an edit distance of 2 , i.e. , it corrects words which have spelling errors that can be corrected by making at most 2 operations of insertion, deletion and substitution of letters in the word. The choice of 2 is governed by the trade off between algorithm runtime and quality of spelling correction. A bigger value improves the spelling correction accuracy but increases the runtime also while a smaller value decreases accuracy and the algorithm runtime. The spelling corrector has been designed as suggested by Peter Norvig[5]. The algorithm requires a dictionary which is used to check if each word of the text exists in it or not. If the word already exists in the dictionary then no change is made to the word and if not, then a candidate list of words is created from the word to be corrected by inserting, substituting or deleting up to 2 letters from it. This list of words is again checked for in the dictionary and returned as suggestions for the word to be corrected. The correction is made with the word formed from lowest edit distance and occurring with more frequency in the dictionary. This makes the edit distance algorithm dependent on the type of dictionary chosen for correction which means the dictionary must be well chosen for spelling correction of a specific document collection. The algorithm runs faster by reading the dictionary only once and keeping a data structure in memory for its word counts which can be referred to whenever a word comes up for correction.

3.3 Spelling Correction Algorithm Evaluation

There has not been much related work regarding automatic evaluation of word-by-word post spelling correction on OCR dataset consisting of Word Split and Join errors. Semi-automatic spelling correction system is discussed in [16] that corrects these errors but requires user interaction in order to perform complete correction and system evaluation. Rice [12] discusses OCR errors similar to the ones in our dataset. Their algorithm evaluates edit distance spelling correction by estimating word accuracy defined as the percentage of correctly recognized words; the length of LCS between correct and incorrect strings on a page by page level is used as the relevant metric. The evaluation strategy works correctly but the definition of accuracy does not give a complete coverage of the spell correction as it does not provide any information on the errors missed by the spelling corrector.

For evaluation on our dataset, the raw OCR text and OCR text after application of spelling correction algorithm needs to be compared with the original newspaper text. The OCR text is extremely garbled with Word Split and Join errors due to which word-to-word alignment with the original newspaper text is impossible. For this purpose, a novel algorithm called SCE (Spelling Correction Evaluation) based on N-gram approach is proposed for automatic evaluation of the corrected text word-by-word against the manually corrected subset of the news articles dataset.

Following sections describe the evaluation parameters for estimating the performance of Spelling Corrector on the OCR dataset used along with the SCE algorithm:

[5]http://norvig.com/spell-correct.html

3.3.1 Evaluation Parameters

•**Accuracy** The evaluation metric used for measuring the performance of Spelling correction algorithm is Accuracy which requires calculation of number of OCR errors that got corrected when compared to the original scanned newspaper text. The measure has been chosen so as to include the complete text coverage and not just check for words that got corrected after spell correction as in the latter case, the number of FP and TN get missed which won't give the correct measure of how well the spell corrector works. The formula used for calculating Accuracy is defined by Manning and Schutze,1999 (p.268-269) as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

where,

TP=Number of True Positives,

TN=Number of True Negatives,

FP=Number of False Positives,

FN=Number of False Negatives.

The aim of the SCE algorithm is to make a word-to-word correspondence between the OCR corrected text and the original OCR text and to mark each token in the OCR text as a TP, FP, TN or FN. Reynaert and Martin [11] suggest a way to define these terms by distinguishing between correct words and incorrect words in the text through the set of non-target, target and selected words and use Precision and Recall evaluation measures for measuring performance of spelling correction.

According to our spelling corrector, a "true positive" is said to occur when a word from the OCR text gets corrected and the corrected spelling matches the one in original article text while a "false positive" occurs if the corrected spelling does not match the corresponding word in the original article text. A "true negative" occurs when a word does not get corrected by the algorithm as it is already correct and matches the correct word in the original text also. On the other hand, a "false negative" occurs when the algorithm is unable to correct the word (there is no change in spelling of the word) and it does not match the corresponding word in the original text but should have been corrected.

•**Time taken for Spelling Correction** Time is also an essential parameter while measuring the performance of spelling correction. Since the dataset is quite large, it is important that the algorithm does not take too long to correct an article and needs to be parallelized in case it takes more time for correction.

•**Person Names Detection Rate(PNDR)** The spelling correction algorithm is evaluated on the basis of another parameter which is used to consider the special case of person entity names spelling correction, as the main goal of research is to detect these names with correct spellings. Person Names Detection Rate can be defined as the ratio of person names recognized through Named Entity Recognition (NER) before spelling correction process and the total number of person names recognized in the original newspaper articles.

$$PersonNamesDetectionRate = \frac{\text{Person Names recognized before/after spelling correction}}{\text{Person Names recognized in original newspaper articles}}$$

3.3.2 SCE (Spelling Correction Evaluation) Algorithm

The SCE algorithm is based on an N-word grams approach. To make the correspondence between corrected and original OCR text, a window of n-word grams in the scanned image text article is considered (Original.txt) which can be seen in a diagrammatic representation in Figure 2.

For each token in the spell corrected text (Corrected.txt), the corresponding token in the scanned text article along with 2 tokens before and 2 tokens after it are considered for alignment[6]. If the token being considered matches with any of these words in the scanned text article words window and its spelling has been corrected when compared to the

[6]The choice of 2 is based on the Word Split and Join errors in the dataset. Choosing n=2 allows a window of words to be considered to make up for the alignment lost because of Word Split and Join errors.

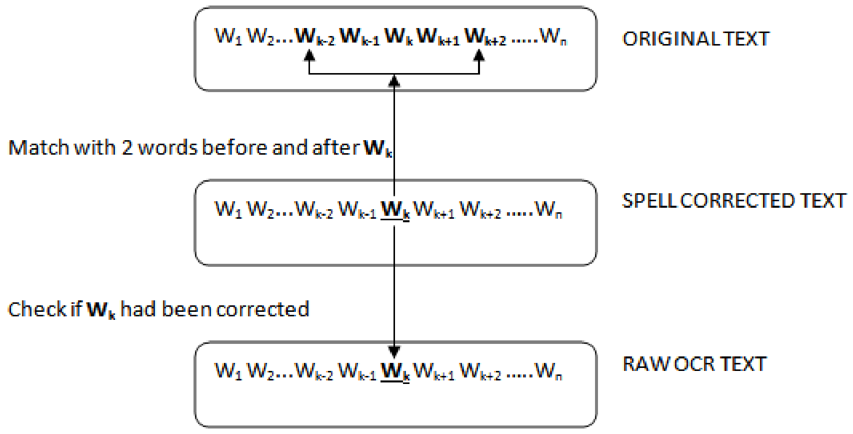

W_1 W_2...W_{k-2} W_{k-1} W_k W_{k+1} W_{k+2} .....W_n
ORIGINAL TEXT
Match with 2 words before and after W_k
W_1 W_2...W_{k-2} W_{k-1} W_k W_{k+1} W_{k+2} .....W_n
SPELL CORRECTED TEXT
Check if W_k had been corrected
W_1 W_2...W_{k-2} W_{k-1} W_k W_{k+1} W_{k+2} .....W_n
RAW OCR TEXT

Figure 2: Schematic diagram for alignment of spell corrected article text with original article text for a word W_k

corresponding token in raw OCR text (OCR.txt), then it is marked as a “True Positive” which is actually rewarding the Spell corrector for making the correct spelling change. A “False Positive” is marked if it does not match any of the words despite its spelling being corrected. If the token being considered matches any of the words in the words window but no spelling correction has been made for it, then it is marked as a “True Negative” and if it does not match any word in the window and the spelling corrector also did not correct it, then it is marked as a “False Negative” as the word got missed by the corrector.

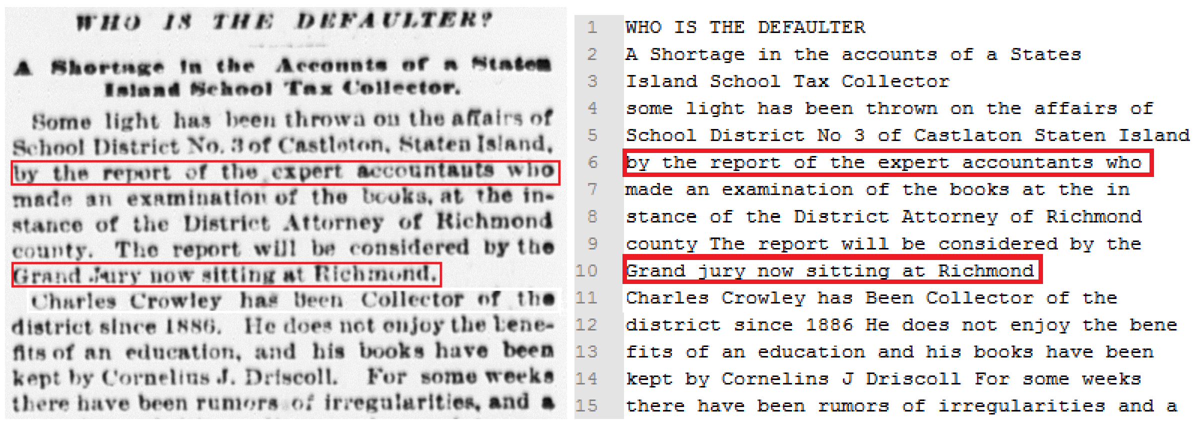
Several cases could occur like difference in the lengths of linetext between OCR and Original text or while considering the first, second or the last tokens from the Corrected text for which the corresponding word window in Original text needs to be smaller. All such cases have been covered in SCE Algorithm 1 which calls function ‘MatchWordGrams’ (Algorithm2) for these different cases.

A limitation of the SCE algorithm is that it requires all 3 versions of a newspaper article (Original, Corrected and OCR) to have the same number of lines. In case of difference in the number of lines of text due to some Word Split and Join errors, the words window needs to be extended so as to cover previous and next line texts also for alignment.

```
Input: Ocr.txt,Corrected.txt,Original.txt
Output: Spell Corrector Accuracy
$OcrLine$:=a line of text from Ocr.txt;
$CorrectedLine$:=a line of text from Corrected.txt;
$OriginalLine$:=a line of text from Original.txt;
$tp \leftarrow 0$ $fp \leftarrow 0$ $tn \leftarrow 0$ $fn \leftarrow 0$;
for (int i=0; i < CorrectedLine.length ; i++) do
    if (CorrectedLine.length<4 || OriginalLine.length<4) then
        MatchWordGrams ((OcrLine,CorrectedLine,OriginalLine,0,OriginalLine.length,i)) ;
    end
    else
        if (i==0) then
            MatchWordGrams ((OcrLine,CorrectedLine,OriginalLine,0,3,0)) ;
        end
        else if  (i==1) then
            MatchWordGrams ((OcrLine,CorrectedLine,OriginalLine,0,4,1)) ;
        end
        else if (i==(CorrectedLine.length-2) || (CorrectedLine.length-1) ||
        (CorrectedLine.length) || (CorrectedLine.length+1)) then
            MatchWordGrams ((OcrLine,CorrectedLine,OriginalLine,i-
            2,OriginalLine.length,i)) ;
        end
        else if (i >= CorrectedLine.length+2) then
            MatchWordGrams ((OcrLine,CorrectedLine,OriginalLine,OriginalLine.length-
            3,OriginalLine.length,i)) ;
        end
        else
            MatchWordGrams ((OcrLine,CorrectedLine,OriginalLine,i-2,i+2,i)) ;
        end
    end
end
$Accuracy = (tp + tn)/(tp + tn + fp + fn)$;
```
Algorithm 1: SCE Algorithm for Spell Correction

```
function MatchWordGrams(OcrLine, CorrectedLine, OriginalLine, jstart, jend, i)
    for (int j=jstart; j<jend; j++) do
        if ((CorrectedLine[i].equals(OriginalLine[j]))&&(!(OcrLine[i].equals(CorrectedLine[i])))) then
            $tp = tp + 1$;
            flag0=false;
            return $tp$;
        end
        else if ((CorrectedLine[i].equals(OriginalLine[j]))&&(OcrLine[i].equals(CorrectedLine[i]))) then
            $tn = tn + 1$;
            flag1=false;
            return $tn$;
        end
    end
    if (!(OcrLine[i].equals(CorrectedLine[i]))&&flag0==true) then
        $fp = fp + 1$;
        return $fp$;
    end
    else if ((OcrLine[i].equals(CorrectedLine[i])) && flag1==true) then
        $fn = fn + 1$;
        return $fn$;
    end
end function
```

Algorithm 2: MatchWordGrams Function called by Algorithm 1



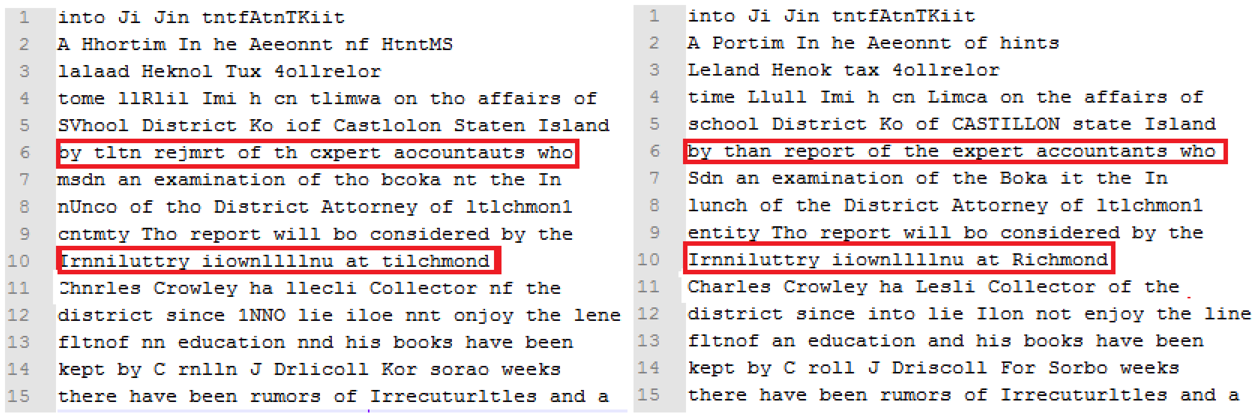
WHO IS THE DEFAULTER?

A Shortage in the Accounts of a Staten Island School Tax Collector.

Some light has been thrown on the affairs of
School District No. 3 of Castleton, Staten Island,
by the report of the expert accountants who
made an examination of the books, at the in-
stance of the District Attorney of Richmond
county. The report will be considered by the
Grand Jury now sitting at Richmond.
Charles Crowley has been Collector of the
district since 1886. He does not enjoy the bene-
fits of an education, and his books have been
kept by Cornelius J. Driscoll. For some weeks
there have been rumors of irregularities, and a

```
1  WHO IS THE DEFAULTER
2  A Shortage in the accounts of a States
3  Island School Tax Collector
4  some light has been thrown on the affairs of
5  School District No 3 of Castlaton Staten Island
6  by the report of the expert accountants who
7  made an examination of the books at the in
8  stance of the District Attorney of Richmond
9  county The report will be considered by the
10 Grand jury now sitting at Richmond
11 Charles Crowley has Been Collector of the
12 district since 1886 He does not enjoy the bene
13 fits of an education and his books have been
14 kept by Cornelins J Driscoll For some weeks
15 there have been rumors of irregularities and a
```

Figure 3: Scanned image of a newspaper article (left) along with its original text (right)



```
1  into Ji Jin tntfAtnTKiit
2  A Hhortim In he Aeeonnt nf HtntMS
3  lalaad Heknol Tux 4ollrelor
4  tome 11Rlil Imi h cn tlimwa on tho affairs of
5  SVhool District Ko iof Castlolon Staten Island
6  by tltn rejmrt of th cxpert aocountauts who
7  msdn an examination of tho bcoka nt the In
8  nUnco of tho District Attorney of 1tlchmon1
9  cntmty Tho report will bo considered by the
10 Irnniluttry iiownllllnu at tilchmond
11 Chnrles Crowley ha llecli Collector nf the
12 district since 1NNO lie iloe nnt onjoy the lene
13 fltnof nn education nnd his books have been
14 kept by C rnlln J Drlicoll Kor sorao weeks
15 there have been rumors of Irrecuturltles and a
```

```
1  into Ji Jin tntfAtnTKiit
2  A Portim In he Aeeonnt of hints
3  Leland Henok tax 4ollrelor
4  time Llull Imi h cn Limca on the affairs of
5  school District Ko of CASTILLON state Island
6  by than report of the expert accountants who
7  Sdn an examination of the Boka it the In
8  lunch of the District Attorney of 1tlchmon1
9  entity Tho report will bo considered by the
10 Irnniluttry iiownllllnu at Richmond
11 Charles Crowley ha Lesli Collector of the
12 district since into lie Ilon not enjoy the line
13 fltnof an education and his books have been
14 kept by C roll J Driscoll For Sorbo weeks
15 there have been rumors of Irrecuturltles and a
```

Figure 4: OCR raw text (left) and Spell corrected text (right) of the article

An Example

Working of the SCE algorithm can be demonstrated with the help of the following example: Consider 3 versions of a scanned image of a newspaper article, the original text of the scanned image in Figure 3 and the raw OCR text and the corrected text (after spell correction) in Figure 4. As highlighted in the figures, for line 6 the line texts are:

OcrLine= *by tltn rejmrt of th cepert aocountauts who*

CorrectedLine= *by than report of the expert accountants who*

OriginalLine= *by the report of the expert accountants who*

Here, for each token of CorrectedLine, we find its index and call the MatchWordGrams function accordingly. For the first token 'by' at index i=0 in CorrectedLine, we consider the word window to be "by the report" (index j=0 to 2) in OriginalLine by matching iteratively with each token to see if there is a match and also if there has been a spelling correction by comparing with the corresponding token in OcrLine. Here, no change was made to the spelling of 'by' and it matches with a word in words window, so it is marked as a FN. For the second token 'than' at index i=1, we consider the word window to be "by the report of" (index j=0 to 3) for which there is no match in the window but there has been a spelling correction from 'tltn' to 'than', which implies the correction was wrong and the token is marked as a FP. For the third token 'report' at index i=2, we consider the window as "by the report of the" (index j=0 to 4) in Original Line and find that there is a match in the word window and there has been a spelling correction too from 'rejmrt' to 'report' which makes this token a TP. Similarly, rest of the tokens get marked for each line in the Corrected.txt.

Another example can be considered from Line 10 in Figure 3 and Figure 4 where the number of tokens is different in CorrectedLine and OriginalLine. In such a case, direct alignment between tokens is not possible because of which the words window becomes useful. Here, when the last token 'Richmond' of CorrectedLine is considered at index i=3, the corresponding words window becomes "Jury now sitting at Richmond" (index j=1 to 5) for which there is a match in the words windows and corresponding spelling has also been changed from 'tilchmond' to 'Richmond' which makes it a TP. Had the word window not been considered, the corresponding token at index j=3 in OriginalLine would have been chosen as 'sitting' which would have resulted in a FP.

3.4 Spelling Correction Algorithm Evaluation Results

Aims: The aim of our experiments is to answer the following questions:

- **Question 1:** How good is the spell corrector? The metrics for evaluation are accuracy and time to correct the text.
- **Question 2:** How good is the Person Names Detection Rate? The metric for evaluation is PNDR.

Materials: The spelling correction algorithm is used to correct all the 14020 OCR raw text articles in the dataset. The dictionary used for look-up is a concatenation of several public domain books from Project Gutenberg and lists of most frequent words from Wiktionary and the British National Corpus[7]. This is augmented with a large people names list which is obtained by running Stanford NER-CRF parser on subsets of the ClueWeb12 dataset made available in the TREC 2013 Crowdsourcing Track[8]. This enhanced dictionary has been used to give special consideration to correction of person names in the dataset.

Methods: In order to answer **Question 1** we do the following:

3 versions of each newspaper article are required: OCR raw text, spelling corrector corrected text and the original scanned newspaper article text. Since the dataset is quite large (14020) and it is not possible to get original text of each of these newspaper images, a smaller number of articles are chosen to study the results of spelling correction. 50 scanned newspaper images are taken and an online OCR [9] is run on them followed by some manual correction to get the original articles text. Accuracy can then be calculated for all 3 versions of 50 newspaper articles using the SCE algorithm by marking each word in the OCR text article as a TP, FP, TN or FN.

In order to answer **Question 2** we do the following:

1. Person Name Detection from raw OCR (**Baseline:**) The NER is run on the raw garbled OCR text.
2. Person Name Detection from spell corrected text (**PND+Spell Correction+Simple Dictionary:**)The NER is run on the spell corrected (using the edit distance algorithm) OCR text without the people names list in the dictionary.
3. Person Name Detection from spell corrected text with enhanced dictionary (**PND+Spell Correction+Enhanced Dictionary:**) The NER is run on the spell corrected (using the edit distance algorithm) OCR text with the enhanced people names list in the dictionary.

Results: The spelling corrector shows an Accuracy of 72.7% when corrected text is compared to OCR text and original article text. We believe that the results are less accurate due to the presence of a large number of Non-word, New Line, Word Split and Join errors in the OCR data which can not be corrected by the spelling correction algorithm used for this research.

The spelling corrector takes 9 seconds on an average to correct the newspaper OCR articles. It takes a total of 36 hours to run on 14020 articles.

But the spelling correction is useful in terms of Person Names Detection Rate *(PNDR)*. Following are the statistics obtained for *PNDR*:

[7]http://norvig.com/big.txt

[8]http://boston.lti.cs.cmu.edu/clueweb12/TRECcrowdsourcing2013/

[9]www.onlineocr.net

PNDR for **Baseline:** : 71.5%

PNDR for **PND+Spell Correction+Simple Dictionary:** : 63.3%

PNDR for **PND+Spell Correction+Enhanced Dictionary:** : 85.5%

These statistics indicate that spelling correction using an extended dictionary for personal names is useful for detecting person names from the garbled newspaper articles and the results are dependent on the type of dictionary being used for spell correction. Table **??** shows some of the person names recognized while calculation of PNDR for each case. The reason for PNDR going down after spell correction with simple dictionary is demonstrated from the table as unnecessary spelling corrections are made while using this dictionary. On the other hand, since the enhanced dictionary contains several person names and no unnecessary spelling corrections are made, more persons name are recognized in this case when compared to the original text leading to a much higher PNDR.

4 Discussion and Future Work

- We believe a better accuracy of spell correction can be obtained by correcting the New Line errors in the articles. This can be done by checking for if the word at last index of a text line or the word at first index of the next text line is a word not present in the dictionary and combining the two and checking again in the dictionary for a valid word. The new word, if present in the dictionary can be replaced by the two words from which it is formed thereby removing the New Line error.
- Similar approach can be applied for Word Split and Join errors but would require each word of an article not present in the dictionary to be analyzed along with some window of words before and after it to make a correction.
- Choice of a dictionary for the edit distance algorithm affects the results as indicated by improvement in PNDR while using an enhanced person names dictionary. We believe using a dictionary with historical terms, places and people names can certainly perform spelling correction better and improve the accuracy also.
- Our spell corrector corrects Non-Real Word errors by focusing on isolated words in the dataset. Other spelling correction algorithms like context dependent spelling correction can also used to correct the dataset and measure accuracy using our SCE algorithm along with other evaluation parameters to compare among multiple algorithms and decide which one suits the dataset better and gives best accuracy.

References

[1] Shivam Agarwal. Utilizing big data in identification and correction of ocr errors. 2013.

[2] Youssef Bassil and Mohammad Alwani. Ocr context-sensitive error correction based on google web 1t 5-gram data set. *arXiv preprint arXiv:1204.0188*, 2012.

[3] Eric Brill and Robert C Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286–293. Association for Computational Linguistics, 2000.

[4] Ishan Chattopadhyaya, Kannappan Sirchabesan, and Krishanu Seal. A fast generative spell corrector based on edit distance. In *Advances in Information Retrieval*, pages 404–410. Springer, 2013.

[5] Peter Christen. A comparison of personal name matching: Techniques and practical issues. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 290–294. IEEE, 2006.

[6] Haimonti Dutta, Rebecca J Passonneau, Austin Lee, Axinia Radeva, Boyi Xie, David L Waltz, and Barbara Taranto. Learning parameters of the k-means algorithm from subjective human annotation. In *FLAIRS Conference*, 2011.

[7] Mohammad Ali Elmi and Martha Evens. Spelling correction using context. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 360–364. Association for Computational Linguistics, 1998.

[8] Karen Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4):377–439, 1992.

[9] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966.

[10] Andres Marzal and Enrique Vidal. Computation of normalized edit distance and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(9):926–932, 1993.

[11] Martin Reynaert. All, and only, the errors: more complete and consistent spelling and ocr-error correction evaluation. In *LREC*, 2008.

[12] Stephen V Rice. *Measuring the accuracy of page-reading systems*. PhD thesis, University of Nevada, 1996.

[13] Christoph Ringlstetter, Max Hadersbeck, Klaus U Schulz, and Stoyan Mihov. Text correction using domain dependent bigram models from web crawls. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2007) Workshop on Analytics for Noisy Unstructured Text Data*, 2007.

[14] Klaus U Schulz and Stoyan Mihov. Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition*, 5(1):67–85, 2002.

[15] Christian M Strohmaier, Christoph Ringlstetter, Klaus U Schulz, and Stoyan Mihov. Lexical postcorrection of ocr-results: The web as a dynamic secondary dictionary? In *ICDAR*, pages 1133–1137. Citeseer, 2003.

[16] Kazem Taghva and Eric Stofsky. Ocrspell: an interactive spelling correction system for ocr errors in text. *International Journal on Document Analysis and Recognition*, 3(3):125–137, 2001.

[17] Xiang Tong and David A Evans. A statistical approach to automatic ocr error correction in context. In *Proceedings of the fourth workshop on very large corpora*, pages 88–100, 1996.

[18] Raghavendra Udupa and Shaishav Kumar. Hashing-based approaches to spelling correction of personal names. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1256–1265. Association for Computational Linguistics, 2010.

[19] Robert A Wagner and Michael J Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.