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
Department of Computer Science
Indraprastha Institute of Information Technology
Delhi, India[1]
aayushee1230@iiitd.ac.in

Haimonti Dutta
Department of Management Science And Systems
University at Buffalo
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 do not 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. 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 Spell Correction Evaluation (SCE) algorithm evaluates its accuracy to be 73.1%. We posit that this novel algorithm for spell correction evaluation has a wide applicability and can play a crucial role in analyzing large volumes of digitized OCR text.

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

1 Introduction

OCR of typed, handwritten or printed text is widely used to obtain digitized text which can be edited, searched, stored and displayed efficiently([11, 18]). It is used in various applications such as banking, digital libraries [10] and repositories, number plate recognition, and handwriting recognition [15]. However, the OCR scanning of printed text generates a lot of garbled text which renders them inadequate for any such tasks. Refinement of such noisy OCR text through spell correction can make them useful for text mining tasks ([5], [20]).

Most spell correction algorithms have focused on improving the correction model and either do not give a detailed performance evaluation of the algorithm post spell correction or the evaluation measures used are not able to completely analyze the performance of such algorithms. A major problem that surfaces when evaluating a spell corrector is that the text has to be verified against the original text (ground truth) to estimate its performance. This one-to-one verification may lead to word alignment problems, since the corrected and original text can be of different lengths. In this paper, we describe the development of an N-word grams Spell Correction Evaluation (SCE) algorithm that can automatically evaluate a spell correction algorithm by using an N-word window to align three parallel corpora - the noisy OCR, corrected and original/ manually cleaned text.

Organization: This paper is organized as follows: related work is described in Section 2; characteristics of the OCR data in Section 3; Spelling Correction and Evaluation algorithms in Section 4; empirical evaluation in Section 5 followed by discussion and future work in Section 6.

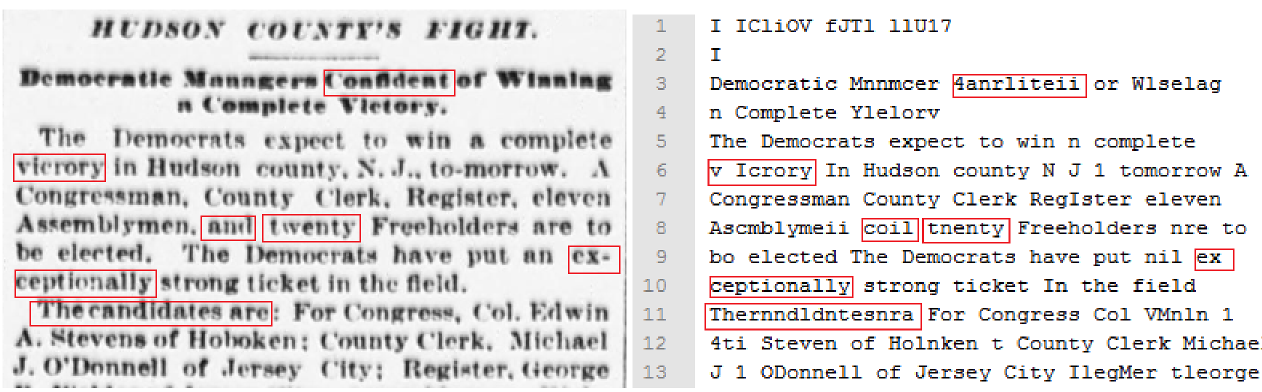
[1]The author is currently affiliated to the Department of Computer Science and Information Technology at Jaypee Institute of Information Technology, Noida, India.

2 Related Work

Kukich [7] 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 ([16], [13]) 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 [6]. 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 Statistical Language Modeling (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]. Agarwal et al. [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.

The 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 [7] [4] which can be corrected using this technique. String edit distance approaches with faster correction are discussed in [9], [14] with variants like Levenshtein automata and normalized edit distance. All of the above algorithms are evaluated based on the percentage of spelling errors corrected or reduction in the word error rate and do not consider the word alignment problem arising due to word split and join errors in the OCR text.

3 Characteristics of OCR data


HUDSON COUNTY'S FIGHT.

Democratic Managers Confident of Winning a Complete Victory.

The Democrats expect to win a complete vierory 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

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

An individual OCR text article has at least one or more of the following types of spelling errors: (1) **Real word errors:** include words that are spelled correctly in the OCR text but still incorrect when compared to the original newspaper article image. Example[2]: "coil" has been correctly spelled in the OCR text but should have been "and" according to the original newspaper article. (2) **Non-real word errors:** include words that have been misspelled due to some insertion, deletion, substitution or transposition of characters from a word. Example: "tnenty" in the OCR text has a substitution error ('n' should have been 'w') which is actually "twenty" according to the original newspaper article. (3) **Non-word errors:** include words that have been spelled incorrectly and are a combination of alphabets and numerical characters. Example: "4anrliteii" which is a combination of alphabets and number and should have been "confident" as per the original newspaper article. (4) **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. Example: "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. (5) **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. Example: "Thernndldntesnra" in the OCR text is actually a

[2] All the examples are illustrated in Figure 1

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.

4 Spelling Correction

The Algorithm The Edit Distance algorithm based on Levenshtein distance [8] has been used for spelling correction. It is an isolated word correction technique that uses dictionary based-look up and the distance between strings for matching the text and correcting it. An “edit distance”[3] corresponds to the minimum number of insertions, deletions, and substitutions required to transform one string into another.

Spelling Correction Algorithm Evaluation For evaluating the performance of spell correction, the raw OCR text and OCR text after application of spelling correction algorithm (corrected text) 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,i.e., the raw OCR and original newspaper text are of different lengths. A novel algorithm,Spelling Correction Evaluation (SCE) based on N-gram approach is proposed for automatic evaluation of the corrected text. The following metrics are used for estimating the performance: (1) **Accuracy** This requires calculation of the number of OCR errors that got corrected when compared to the original newspaper text. Specifically, $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$ where, TP=Number of True Positives, TN=Number of True Negatives, FP=Number of False Positives, FN=Number of False Negatives. Reynaert and Martin [12] 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 which we adapt for this work. (2) **Time taken for Spelling Correction** The time for correcting the text is also noted for benchmarking correction of large datasets.

N-Word Grams Spelling Correction Evaluation(SCE) Algorithm To make the correspondence between corrected and original OCR text, a window of N-word grams in the original newspaper text is considered which can be seen in a diagrammatic representation in Figure 2. For each token in the spell corrected text, the corresponding token in

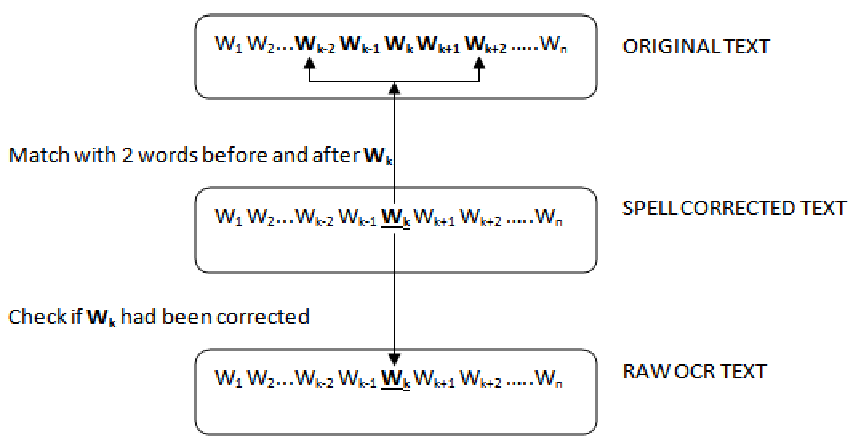


Figure 2: Schematic diagram for alignment of spell corrected article text with original article text for a word W_k

the original text article along with 2 tokens before and 2 tokens after it are considered for alignment[4]. If the token

[3]Our edit distance algorithm corrects non-real word spelling errors 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. The spelling corrector has been designed as suggested by Peter Norvig `http://norvig.com/spell-correct.html`.

[4]The choice of N=2 is based on the Word Split and Join errors in the dataset. This value can be set appropriately by considering the maximum difference of lengths in each line of OCR and original text in the dataset.

being considered matches with any word in the original text article word's window and its spelling has been corrected when compared to the corresponding token in raw OCR text, 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. Table 1 describes the the process of marking each token in the corrected text as a TP, TN, FP or FN in each text article for calculation of accuracy. The final values of TP, TN, FP and FN are accumulated throughout the dataset to calculate accuracy.

Table 1: Calculation of accuracy in SCE algorithm

Criteria to be checked \ Evaluation Metric	TP	TN	FP	FN
Match found in N-word window?	Yes	Yes	No	No
Spell correction was done?	Yes	No	Yes	No

Several scenarios could arise during the word alignment process due to difference in the lengths of text between OCR and original text. All such cases are depicted in Table 2 which describes the window size of tokens to match in the original text (from j=starting index to ending index) for every token *i* of the corrected text.

Table 2: Different cases for word alignment

Token index of CorrectedLine(i) \ Token index of OriginalLine	Starting index (j)	Ending index (j)
Length[CorrectedLine] <4 or Length[OriginalLine]<4	0	Length[OriginalLine]
i=0	0	3
i=1	0	4
i=Length[CorrectedLine]-2	i-2	Length[OriginalLine]
i=Length[CorrectedLine]-1	i-2	Length[OriginalLine]
i=Length[CorrectedLine]	i-2	Length[OriginalLine]
i=Length[CorrectedLine+1]	i-2	Length[OriginalLine]
i>=Length[CorrectedLine]+2	Length[OriginalLine]-3	Length[OriginalLine]
Any other value of i	i-2	i+3

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 as alignment of line texts is performed. In case of difference in the number of lines of text due to some Word Split and Join errors, the word's window needs to be extended so as to cover previous and next line texts also for alignment.

5 Empirical Evaluation

Data Source The dataset used for empirical evaluation of the algorithm has been obtained from the Chronicling America[5] website. It contains scanned newspaper pages published in New York between1890 to 1920. OCR software is run over high resolution images to create searchable full text of the newspaper articles.

Data Statistics 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 are used for empirical evaluation. The text does not have any punctuation and contains a large amount of garbled text containing OCR errors mentioned in Section 3.

Experimental Procedure and Results

[5] `http://chroniclingamerica.loc.gov/`

Aim: The aim of our experiments is to answer the following question: How good is the spell corrector? The metrics for evaluation are accuracy and time taken to correct the text.

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[6]. 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[7]. This enhanced dictionary has been used to give special consideration to correction of person names in the dataset.

Methods: In order to answer our research question for checking the performance of spell corrector, 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 sample of articles is chosen to study the results of spelling correction. 50 scanned newspaper images are taken and an online OCR[8] 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.

Results: The spell 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. The spell corrector also shows an Accuracy of 73.1% when corrected text is compared to OCR text and original article text using our SCE algorithm. 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 edit distance spelling corrector used for this research.

Discussion We compared our N-gram based SCE algorithm with the LCS (Longest Common Subsequence) algorithm[9]. The LCS of corrected and original text gives a list of matching corrected words found in the original text. Following the similar evaluation procedure of calculating accuracy as in the N-word gram approach, it was found that there is no statistically significant difference in accuracy when using either of the two algorithms. We posit that LCS is a special case of the N-word gram algorithm when the window size N is set to the complete text in a line.

6 Conclusion and Future Work

In this paper, we presented a novel approach for automatic performance evaluation of spell correction on noisy OCR text through N-word grams alignment of the OCR, corrected and manually cleaned text. Preliminary results of application of our algorithm on an Edit distance based spell corrector evaluate its accuracy to be 73.1%. In future, we plan to use other spelling correction algorithms like context dependent spelling correction to correct the OCR text and measure the accuracy using our SCE algorithm.

7 Acknowledgement

This work has been supported by the National Endowment of Humanities Grant, NEH HD-51153-10.

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.

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

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

[8] www.onlineocr.net

[9] https://en.wikipedia.org/wiki/Longest_common_subsequence_problem

[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] 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.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] Thomas McMurdo and Birdie MacLennan. The vermont digital newspaper project and the national digital newspaper program. *Library Resources & Technical Services*, 57(3):148–163, 2013.

[11] Thomas Palfray, David Hebert, Stéphane Nicolas, Pierrick Tranouez, and Thierry Paquet. Logical segmentation for article extraction in digitized old newspapers. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 129–132. ACM, 2012.

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

[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] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. A survey of ocr applications. *International Journal of Machine Learning and Computing*, 2(3):314–318, 2012.

[16] 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.

[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] Andrew J Torget, Rada Mihalcea, Jon Christensen, and Geoff McGhee. Mapping texts: Combining text-mining and geo-visualization to unlock the research potential of historical newspapers. 2011.

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

[20] Tze-I Yang, Andrew J Torget, and Rada Mihalcea. Topic modeling on historical newspapers. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 96–104. Association for Computational Linguistics, 2011.