Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing

Thi-Tuyet-Hai Nguyen
L3i, University of La Rochelle
La Rochelle, France
hai.nguyen@univ-lr.fr

Adam Jatowt
Graduate School of Informatics,
Kyoto University
Kyoto, Japan
adam@dl.kuis.kyoto-u.ac.jp

Mickael Coustaty
L3i, University of La Rochelle
La Rochelle, France
mickael.coustaty@univ-lr.fr

Nhu-Van Nguyen
L3i, University of La Rochelle
La Rochelle, France
nhu-van.nguyen@univ-lr.fr

Antoine Doucet
L3i, University of La Rochelle
La Rochelle, France
antoine.doucet@univ-lr.fr

ABSTRACT

Post-OCR is an important processing step that follows optical character recognition (OCR) and is meant to improve the quality of OCR documents by detecting and correcting residual errors. This paper describes the results of a statistical analysis of OCR errors on four document collections. Five aspects related to general OCR errors are studied and compared with human-generated misspellings, including edit operations, length effects, erroneous character positions, real-word vs. non-word errors, and word boundaries. Based on the observations from the analysis we give several suggestions related to the design and implementation of effective OCR post-processing approaches.

KEYWORDS

OCR errors, OCR post-processing, post-OCR text correction

1 INTRODUCTION

In an effort to preserve and provide an easy access to past documents, optical character recognition (OCR) techniques have been developed to transform paper-based documents into digital documents. However, various layouts and poor physical quality of degraded documents pose big challenges to OCR engines. Post-OCR is crucial for improving the quality of OCR documents by detecting and correcting errors.

Although OCR errors share some common features with spelling errors, OCR errors have their own special characteristics as they are created by different processes than spelling errors. Naturally, better understanding of OCR errors can help to create better post-OCR approaches. However, to this date, few analyses were done to uncover common characteristics of OCR errors, and they all have been on a coarse level [13, 23]. This paper reports then the results of the analyses of various characteristics of OCR errors on popular public datasets, and compares them with misspellings. Particularly, edit operation types and edit distance are considered. In addition, we concentrate not only on word lengths but also on OCR token lengths. Moreover, positions of incorrect characters and *real-word* vs. *non-word* errors are analyzed. Problems related to the wrong deletion/insertion of white spaces (word boundaries) are also examined.

For the analysis, we utilize four public English datasets along with their ground truth data. Two of them come from the English part of the Post-OCR text correction competition dataset [4] - the largest public, aligned dataset of this kind [5] [1]. Two others are the OverProof Evaluation data [7][2]. While other datasets contain synthetic data or are private, these two datasets (and their manual GT) include OCR texts of old documents collected from two well-known libraries and are made public.
Our analysis should be beneficial for researchers and practitioners helping them better understand strengths as well as weakness of their approaches. Based on the reported results, we also provide guidelines for building more effective post-processing approaches.

To sum up, we make the following contributions in this paper.

(1) Firstly, we analyze OCR errors and compare them with human-generated misspellings in several aspects. Our analysis forms the basis for better judgment of the pros and cons of post-OCR approaches and for improving their performances.
(2) Secondly, we also make statistics on some extended aspects beside typical ones for spelling errors characterization [13], such as string similarities between errors and their ground truth words based on Longest Common Sequence (LCS), OCR token lengths and different erroneous character positions.

[1] https://sites.google.com/view/icdar2017-postcorrectionocr/
[2] http://overproof.projectcomputing.com/datasets/

978-1-7281-1547-4/19/$31.00 ©2019 IEEE
DOI 10.1109/JCDL.2019.00015

(3) To provide clearer views about OCR errors, novel error type classifications are proposed. In particular, we review challenges of correcting *short-word*/*long-word* errors with large/small edit distances by grouping errors according to word length. In addition, *real-word*/*non-word* errors are also categorized according to word-boundary problem.
(4) Finally, based on our observations, we recommend several suggestions for designing OCR post-processing techniques, such as ones related to edit distance thresholds, frequent edit operation types, erroneous character positions, etc.

The remainder of this paper is organized as follows. We introduces four datasets we work on in Sec. 2. Then, Sec. 3 surveys related work. In Section 4, we analyze OCR errors and give many useful statistics. After that, the summary of our major findings is shown in Section 5. Finally, conclusions are discussed in Section 6.

2 DATASETS

Four analyzed datasets are public collections of historical documents obtained from four libraries.

Two first datasets come from ICDAR2017 Post-OCR text correction competition [4]. The competition data contains OCR processed text of ancient English and French documents from two national libraries, the National Library of France (BnF) and the British Library (BL). The corresponding ground truth (GT) was created by different projects (such as Gutenberg, Europeana Newspapers). In this paper, we focus on English OCR text of this multilingual dataset which consists of 813 files belonging to two types: monograph and periodical. The competition organizers divided this English OCR text into two datasets, Monograph and Periodical. There is no information about which OCR engines were used to generate the OCR text of the competition dataset.

Two others are Overproof evaluation datasets [7]. The first one (denoted as OverNLA) consists of 159 medium-length news articles with at least 85% correct lines, which were extracted from one of the longest-running titles in the National Library of Australia's Trove newspaper archive - The Sydney Morning Herald, 1842-1954. Its corresponding GT was additionally corrected by Evershed *et al.* [7] after crowd sourcing corrections [8]. The second one (denoted as Overproof LC) consists of 49 medium-length news articles randomly selected from 5 titles of the Library of Congress Chronicling America newspaper archive. The corresponding GT of OverNC was manually corrected by Evershed *et al.* [7]. Both of the Overproof datasets are noisier than the competition ones. Their combined size is 208 articles/files, and they were processed by ABBYY FineReader[3], which is the state-of-the-art commercial OCR system.

The four datasets thus contain OCR texts of past documents from popular libraries (National Library of France, British Library, National Library of Australia, Library of Congress Chronicling America). The included documents are characterized by varying levels of degradation under independent conservation and originate from a relatively wide time range spanning from 1744 to 1954. In view of these, altogether the datasets are representative for historical OCR texts with typical OCR errors. The details of sources, types, years, word error rates (W.E.R), sizes and the file counts of all the four datasets are listed in Table 1.

Table 1: Sources, types, years, word error rates (W.E.R), sizes and a number of files of four datasets.

Sources	Types	Years	W.E.R.	Sizes	Files
Monograph	monograph	1862-1911	9%	4.2M	747
Periodical	periodical	1744-1894	16%	1.8M	66
OverNLA	news	1842-1954	25%	0.3M	159
OverLC	news	1871-1921	27%	0.1M	49

3 RELATED WORK

This paper studies OCR errors and compares them with human-generated misspellings. Our observations are then used for drawing several suggestions towards designing OCR post-processing methods. Consequently, in the two following sections, we review works related to misspellings, OCR errors and post-OCR approaches.

3.1 Misspellings and OCR errors

Due to certain shared features between misspellings and OCR errors, an overview of misspelled words could give basic ideas on OCR errors. Kukich [13] made a coarse-grained survey on spelling error characteristics and automatic spellers. Similar features of misspellings were described in [22, 23]. Spelling errors have been studied from the viewpoint of basic edit operation types, word length effects, *first-position* errors, *non-word*/*real-word* errors, and word boundaries.

Firstly, depending on edit distance, there are *single-error* tokens with edit distance of 1 (e.g. *'school'* vs. *'schopl'*) and *multi-error* tokens with higher edit distance (e.g. *'school'* vs. *'schopi'*). Damerau [6] and Mitton [17] indicated that *single-error* typos were around 80%, 69% of misspellings, respectively. Thus, the average rate of *single-error* typos can be considered as 74.5%.

Secondly, word lengths have been also considered from the viewpoint of misspellings tendency. Errors were examined as for whether they appear in short words (defined as words of 2, 3 and 4 characters) or longer-length words. Let us call errors involving short words as *short-word* errors. Kukich [12] found that 63% of errors involved short words.

Thirdly, misspellings can occur at the first character (e.g. *'world'* vs. *'uorld'*) or at other characters (e.g. *'world'* vs. *'workd'*, *'world'* vs. *'worlh'*). Mitton [17] described that 7% of the misspellings of his dataset appeared at the first character. In the dataset of Kukich [12], that proportion was 15%. The average rate of *first-position* errors can be then considered to be around 11% of misspellings.

Next, if a token is not a lexicon entry, it is deemed a *non-word* error. In this case, determination of an error depends then on the coverage and quality of a particular lexicon used. If a valid word occurs in a wrong context, it is considered as a *real-word* error. For example, in two phrases *'glow-worm candles'* vs. *'glow-wonn candies'*, a *non-word* error is *'glow-wonn'* and *'candies'* is a *real-word* error. Researches on different datasets informed different rates of *real-word* errors. Mitton [17] revealed that 40% of misspelled words involved *real-word* errors. Young *et al.* [28] showed that the rate of *real-word* errors of their corpus was 25%. On average, one could assume that 67.5% of misspellings are related to *non-word* errors.

As to the problem of word boundaries, wrongly deleting/inserting white spaces results in *incorrect split* errors (e.g. *'depend'* vs. *'de*

[3] https://www.abbyy.com

pend') and *run-on* errors (e.g. *'is said'* vs. *'issaid'*). In the corpus of Kukich [12], the percent of word boundary spelling errors were 15% with 13% of *run-on* errors, and 2% of *incorrect split* errors. Moreover, Kukich mentioned that OCR text tended to split than to join tokens.

While Kukich mainly focused on spelling errors, Nagy *et al.* [18] concentrated on examining selected examples of erroneous OCR tokens. Their work pointed out possible causes of OCR errors, including imaging defects, similar symbols, punctuation, and typography, then gave several potential solutions. However, it did not provide any detailed statistics on each source of OCR errors.

3.2 OCR post-processing approaches

A typical post-processing approach consists of two steps, detecting and correcting errors. In terms of the detection task, dictionary and character n-gram models are often used to detect *non-word* errors. In terms of the correction task, for each OCR error, the list of candidates are generated based on different sources at character level, word level. The best candidate is the correction in an automatic mode, or the top *n* candidates are suggested to correct the error in a semi-automatic mode.

A wide range of approaches was devoted to OCR post-processing, which can be classified into two main types: dictionary-based and context-based types. The *dictionary-based type* aims to correct isolated-word errors and does not take nearby context into consideration [3, 20], hence this type cannot deal with *real-word* errors. The *context-based type*, which considers grammatical and semantic contexts of errors, promises to overcome the issues of the first type. Most of the techniques of this type rely on noisy channel and language model [1, 15, 27]. The others explore several machine learning techniques to suggest correct candidates [2, 10, 16].

Jones *et al.* [1] and Tong *et al.* [27] explored several features, including character n-grams, character confusion (or device mapping statistics), and word bi-gram in different ways to detect and correct erroneous OCR tokens. Using similar features, Llobet *et al.* [15] built an error model and a language model, then added one more model built from character recognition confidences, called hypothesis model. Three models were compiled separately into Weighted Finite-State Transducers (WFSTs), then were composed into the final transducer. The best token was the lowest cost path of this final transducer. However, character recognition confidence is often missing at least with the whole competition dataset [4] and Overproof evaluation datasets [7].

Along with the development of machine translation techniques, some approaches considered OCR post-processing as machine translation (MT), which translates OCR text into the correct one in the same language. Afli *et al.* [2] and some competition approaches of the competition [4] applied machine translation techniques (from statistical MT, neural MT to hybrid MT at word and/or character level) to deal with detecting and correcting OCR errors.

Other approaches [10, 16] explored different sources to generate candidates and then ranked them using a regression model. Several features were extracted such as confusion probability, uni-gram frequency, context feature, term frequency in the OCR text, word confidence, and string similarity. Then, a regression model was used to predict the best candidate for erroneous OCR token.

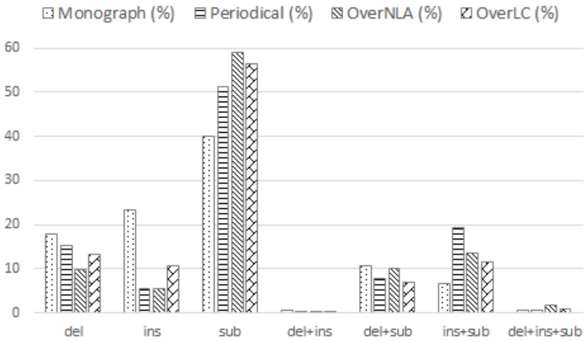

Figure 1: Error rates based on edit operation types

Post-processing approaches offered different views about OCR errors however none of them gave a general hierarchy of OCR errors. Some mainly focused on *real-word* and *non-word* errors [27]. Other approaches considered errors with segmentation at word or character level [10, 16]. Lastly, some others [1, 2, 7, 24–26] just gave examples of OCR errors without any detailed statistics.

In contrast to the above-discussed researches, our work focuses on analyzing OCR errors and gives detailed statistics based on four public datasets. Besides the aspects mentioned in the survey [13], we examine additional features like non-standard substitution mappings, different erroneous character positions, OCR token lengths. Moreover, we give novel classifications and provide several suggestions about the design of post-OCR techniques.

4 ANALYSIS OF OCR ERRORS

In the following sections, we present five main types of analyses conducted on all the datasets.

4.1 Edit operations

In this section, we discuss edit operation types, standard/non-standard substitution mappings (denoted as standard/non-standard mappings), edit distance and string similarity based on LCS.

4.1.1 Edit operation types. In order to transform token A to token B, four basic edit operation types can be performed: deletion, insertion, substitution, and transposition [6]. Prior works [11, 16, 27] indicated that transposition is common in misspellings but rarely occurs in OCR errors. We then only consider the three first types.

Fig. 1 shows the percentages of single modification error types (deletion, insertion and substitution denoted as *del*, *ins* and *sub*, respectively) and ones of their possible combinations (*del+ins, del+sub, ins+sub, del+ins+sub*, respectively) in all the four datasets. Among single edit operation types, the average percentage of substitution (51.6%) is much higher than that of two others. Furthermore, the total percentage of three single edit operation types is about 77.02%, thus higher than that of their combinations. It leads to the conclusion that post-OCR techniques can correct most of errors by just concentrating on a single modification type.

As to the combinations of edit operation types, deletion and insertion rarely occur together. In fact, the combinations of deletion and insertion have very small occurrence rate being 0.24%

(*del+ins*) and 1% (*del+ins+sub*). Post-OCR approaches could then in our opinion pay less attention on the combinations in candidate generation.

Moreover, the average rate of OCR errors involving substitution, insertion, deletion are approximately 5:1:1, which is useful information for some post-OCR approaches [7, 15, 19] to decide the number of substitution/insertion character candidates for each OCR character position in candidate generation. If the rate is too small, no correct candidates can be suggested. Otherwise, many incorrect candidates are created negatively affecting the candidate ranking process.

4.1.2 Standard mapping. Secondly, we consider standard and non-standard mappings. While misspellings often have standard mapping 1:1 (e.g. *'hear'* vs. *'jear'*), OCR errors contain not only standard mappings 1:1 but also non-standard mappings, such as *n*:1 and 1:*n* (e.g. *'link'* vs. *'hnk'*, *'link'* vs. *'liiik'*).

The standard mapping 1:1 of our datasets is illustrated in Table 2. In this table, we compute the percentage of appearance frequency of each GT character being recognized as an OCR character for each dataset. Let us name this percentage as mapping percentage. In order to make the table compact, we only show OCR characters whose mapping percentages are more than 0.1%. Other cases whose mapping percentages are less than 0.1% are denoted as @. Because 1 GT character can be recognized as 1 or *n* OCR characters, so other cases include OCR characters in 1:1 mappings and 1:*n* mappings. For example, the percentages of frequency of character *b* in Periodical being recognized as *'b'*, *'h'* and other characters are 96.7%, 1.6% and 1.7%, respectively.

Table 2 indidates that the characters with the highest and lowest recognition accuracy are *t, z* with 98.53% and 88%, respectively. Moreover, the statistics also reveal that characters sharing similar shapes are easily confused, such as *'b' vs. 'h'; 'c' vs. {'o', 'e'}; 'e' vs. {'o', 'c'}.*

This standard mapping is used to create character confusion matrix - one of the most important sources to generate and rank candidates. It is obvious that the more similar frequent error patterns between a training part and a testing part of the used datasets are, the higher the probability that the correct candidates are generated. However, OCR errors can vary from OCR engines, layouts as well as degradation levels of documents, and etc. Therefore, some very frequent characters along with their highly possible misrecognition (e.g. *'e'* vs. *'o'*, *'j'* vs. *'i'*) may not occur in the large training part and only appear in the small testing part. In such cases, it is impossible to generate valid candidates for unseen error patterns of the testing part.

4.1.3 Non-standard mappings. Besides the standard mapping 1:1, OCR errors are also subject to more complex mappings [1, 12]. Different from past related work, our study provides the detailed statistics on the four popular datasets instead of only giving examples of non-standard mappings.

The first point is 1:*n* mapping, in which one GT character is recognized as *n* OCR characters (e.g. *'main'* vs. *'rnain'*). The mapping percentages of frequency of each GT character being recognized as *n* OCR characters are calculated for each dataset in Table 3. With the same compactness reason as in Table 2, this table only contains *n* OCR characters whose mapping rates are greater than 0.01%. As mentioned in Sec 4.1.2, in Table 2, character @ denotes other characters of 1:1 and 1:*n* mappings. Table 3 clarifies the 1:*n* mapping. For instance, the percentage of frequency of character *'b'* in Periodical being recognized as *'li'*, *'ti'*, *'th'*, *'l.'* are 0.19%, 0.02%, 0.02%, 0.02%, respectively. The 1:*n* mapping statistics indicate that there are some frequent patterns along with their average percents, such as *'b'*{*'li'*:0.05, *'h'*:0.03}; *'d'*{*'il'*:0.07, *'cl'*:0.03}; *'h'*{*'li'*:0.34, *'ii'*:0.06}.

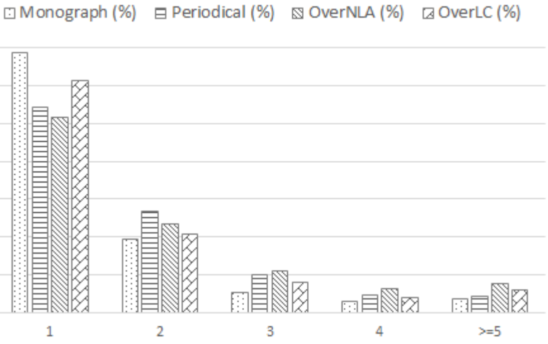

Figure 2: Error rates based on edit distances

The second point is *n*:1 mapping, in which *n* GT characters are recognized as one OCR character (e.g. *'main'* vs. *'mam'*). The frequency rates of *n* GT characters being recognized as one OCR character are computed on four datasets in Table 4. This table only shows GT character ngrams whose mapping percentages are higher than 0.01% and which appear at least 10% of max frequency of their ngrams. Different from Table 2 and 3, in Table 4 we group percentages according to OCR characters because it is inefficient to show many GT character ngrams in the first column. For example, in Monograph dataset, the percentage of appearance frequency of GT character bigram *'li'* being recognized as *'b'* is 0.03%.

Based on the statistics of *n*:1 mappings, some common patterns with their average rates emerge, (shown as 1 OCR character: *n* GT characters), such as *'b'*{*'si'*:0.05, *'li'*:0.04}; *'d'*{*'il'*:0.7, *'ll'*:0.12}; *'h'*{*'li'*:0.16, *'ly'*:0.1}.

Our observations on these mappings support a conclusion that some characters *'b'*, *'d'*, *'h'*, *'m'*, *'n'* are easily recognized as *'li'*, {*'il'*, *'cl'*}, *'li'*, {*'rn'*, *'in'*}, {*'ri'*, *'ii'*}, respectively. In opposite way, *'li'*, {*'il'*, *'cl'*}, *'li'*, {*'rn'*, *'in'*}, {*'ri'*, *'ii'*} can be recognized as *'b'*, *'d'*, *'h'*, *'m'*, *'n'*, respectively. These kinds of mappings also play important roles in generating and ranking candidates.

It should be noted that the statistics of these non-standard mappings are extracted from aligned OCR and their corresponding GT. Although we make a full use of OCR text along with its corresponding GT, there are still some unavoidable noises in our statistics due to the lack of character recognition confidences from OCR engines.

4.1.4 Edit distances. In case of edit distances, the survey on spelling errors [13] pointed out two main types: *single-error* tokens (with one edit distance) and *multi-error* tokens (with higher edit distances). It is obvious that the smaller edit distance an error has, the easier the correction task is.

Percentages of errors based on edit distances of our datasets in Fig. 2 show that most of OCR errors are *single-error* tokens with approximately 58.92% occurrences. That rate is smaller than the rate

Table 2: Percentages of standard mapping 1:1 (one GT character is substituted by one OCR character). Only values higher than 0.1% are shown, other characters (including sequences of more than one character) are denoted as @.

GT Char	Monograph	Periodical	Overproof NLA	Overproof LC
a	{a: 99.5, @: 0.5}	{a: 97.5, u: 0.4, n: 0.2, e: 0.2, i: 0.2, @: 1.5}	{a: 92.7, n: 2.1, i: 1.1, u: 1.0, o: 0.3, m: 0.2, @: 2.6}	{a: 92.8, n: 2.7, u: 0.8, i: 0.5, m: 0.3, o: 0.2, @: 2.7}
b	{b: 98.7, h: 0.8, @: 0.5}	{b: 96.7, h: 1.6, @: 1.7}	{b: 96.2, h: 1.7, l: 0.5, t: 0.3, @: 1.3}	{b: 93.9, h: 1.8, l: 0.5, n: 0.4, i: 0.3, t: 0.3, o: 0.2, m: 0.2, @: 2.4}
c	{c: 97.0, o: 2.0, e: 0.6, @: 0.4}	{c: 96.2, e: 1.2, o: 1.0, @: 1.6}	{c: 93.9, e: 1.7, o: 1.5, r: 0.4, t: 0.2, i: 0.2, @: 2.1}	{c: 92.2, o: 3.1, e: 1.6, u: 0.3, s: 0.3, n: 0.2, a: 0.2, r: 0.2, t: 0.2, @: 1.7}
d	{d: 99.7, @: 0.3}	{d: 98.4, l: 0.2, i: 0.2, @: 1.2}	{d: 97.1, a: 0.4, l: 0.2, i: 0.2, @: 2.1}	{d: 96.8, l: 0.5, i: 0.3, u: 0.3, @: 2.1}
e	{e: 98.7, o: 0.2, @: 1.1}	{e: 96.9, o: 0.6, c: 0.5, a: 0.2, s: 0.2, @: 1.6}	{e: 86.1, o: 9.2, c: 1.6, i: 0.3, a: 0.2, @: 2.6}	{e: 80.8, o: 14.8, c: 0.9, u: 0.4, i: 0.3, r: 0.2, n: 0.2, @: 2.4}
f	{f: 98.2, @: 1.8}	{f: 96.2, t: 1.2, l: 0.9, i: 0.4, @: 1.3}	{f: 94.3, l: 1.5, t: 1.0, i: 0.9, @: 2.3}	{f: 94.1, l: 1.8, t: 1.4, i: 0.6, @: 2.1}
g	{g: 99.6, @: 0.4}	{g: 98.3, @: 1.7}	{g: 93.4, c: 0.4, p: 0.4, r: 0.4, e: 0.3, s: 0.3, i: 0.3, u: 0.3, t: 0.2, f: 0.2, @: 3.8}	{g: 95.2, j: 0.3, i: 0.3, c: 0.2, e: 0.2, @: 3.8}
h	{h: 99.1, b: 0.4, @: 0.5}	{h: 95.2, b: 1.7, i: 0.4, n: 0.2, @: 2.5}	{h: 95.1, b: 1.1, l: 0.8, i: 0.7, n: 0.2, @: 2.1}	{h: 95.7, l: 1.0, i: 0.6, b: 0.5, n: 0.3, @: 1.9}
i	{i: 99.1, @: 0.9}	{i: 97.6, l: 0.6, t: 0.2, @: 1.6}	{i: 90.7, l: 3.3, m: 0.4, t: 0.3, u: 0.2, n: 0.2, @: 4.9}	{i: 94.0, l: 1.6, @: 4.4}
j	{j: 99.7, @: 0.3}	{j: 97.4, i: 0.3, l: 0.3, c: 0.2, @: 1.8}	{j: 85.0, i: 1.5, l: 0.4, t: 0.4, @: 12.7}	{j: 92.7, @: 7.3}
k	{k: 99.5, @: 0.5}	{k: 98.6, t: 0.2, @: 1.2}	{k: 95.6, l: 1.0, i: 0.3, h: 0.2, t: 0.2, @: 2.7}	{k: 97.5, a: 0.2, i: 0.2, h: 0.2, @: 1.9}
l	{l: 95.6, i: 0.8, d: 0.2, @: 3.4}	{l: 96.9, i: 0.8, t: 0.2, @: 2.1}	{l: 96.2, i: 0.8, @: 3.0}	{l: 96.8, i: 0.7, @: 2.5}
m	{m: 99.1, @: 0.9}	{m: 97.4, n: 0.5, i: 0.2, @: 1.9}	{m: 94.3, n: 1.6, i: 0.8, r: 0.5, u: 0.2, @: 2.6}	{m: 93.9, n: 1.3, i: 1.1, u: 0.3, r: 0.2, t: 0.2, @: 3.0}
n	{n: 99.1, u: 0.2, @: 0.7}	{n: 96.4, u: 1.2, a: 0.3, m: 0.2, o: 0.2, i: 0.2, @: 1.5}	{n: 96.2, u: 1.0, i: 0.4, m: 0.3, a: 0.2, @: 1.9}	{n: 92.6, u: 4.0, i: 0.8, m: 0.2, a: 0.2, @: 2.2}
o	{o: 99.4, @: 0.6}	{o: 97.9, e: 0.5, a: 0.2, @: 1.4}	{o: 98.0, n: 0.2, i: 0.2, @: 1.6}	{o: 97.2, n: 0.3, u: 0.3, e: 0.3, @: 1.9}
p	{p: 99.8, @: 0.2}	{p: 98.7, n: 0.2, @: 1.1}	{p: 97.9, n: 0.7, i: 0.2, r: 0.2, @: 1.0}	{p: 96.8, n: 0.5, j: 0.3, o: 0.2, i: 0.2, r: 0.2, @: 1.8}
q	{q: 99.4, @: 0.6}	{q: 97.7, o: 0.2, i: 0.2, j: 0.2, @: 1.7}	{q: 97.3, a: 1.5, o: 0.9, @: 0.3}	{q: 90.7, i: 3.3, m: 2.9, @: 3.1}
r	{r: 99.4, @: 0.6}	{r: 98.5, i: 0.3, t: 0.2, @: 1.0}	{r: 93.4, i: 3.3, l: 0.4, n: 0.3, t: 0.2, @: 2.4}	{r: 98.1, i: 0.2, t: 0.2, @: 1.5}
s	{s: 98.8, a: 0.5, f: 0.3, @: 0.4}	{s: 94.2, a: 0.8, e: 0.7, t: 0.3, i: 0.3, @: 3.7}	{s: 91.7, a: 1.2, i: 0.5, e: 0.3, n: 0.2, b: 0.2, t: 0.2, @: 5.7}	{s: 90.8, t: 0.6, i: 0.5, e: 0.5, a: 0.4, n: 0.3, f: 0.3, u: 0.2, l: 0.2, o: 0.2, h: 0.2, @: 5.8}
t	{t: 99.7, @: 0.3}	{t: 98.7, i: 0.2, l: 0.2, @: 0.9}	{t: 97.7, l: 0.7, i: 0.2, @: 1.4}	{t: 98.0, l: 0.6, i: 0.2, @: 1.2}
u	{u: 99.2, n: 0.2, @: 0.6}	{u: 96.6, n: 1.1, a: 0.7, o: 0.3, i: 0.2, @: 1.1}	{u: 96.1, n: 1.0, i: 0.6, a: 0.3, m: 0.2, @: 1.8}	{u: 96.1, i: 0.7, a: 0.5, o: 0.2, n: 0.2, j: 0.2, @: 2.1}
v	{v: 99.6, @: 0.4}	{v: 97.9, r: 0.7, y: 0.2, @: 1.2}	{v: 92.2, i: 0.8, r: 0.5, y: 0.3, n: 0.3, t: 0.2, @: 5.7}	{v: 97.7, i: 0.3, r: 0.3, m: 0.3, @: 1.4}
w	{w: 99.6, @: 0.4}	{w: 98.7, @: 1.3}	{w: 92.8, v: 1.1, n: 0.5, y: 0.3, m: 0.2, i: 0.2, @: 4.9}	{w: 98.1, v: 0.2, o: 0.2, @: 1.5}
x	{x: 99.0, @: 1.0}	{x: 97.4, i: 0.8, s: 0.2, r: 0.2, t: 0.2, @: 1.2}	{x: 94.6, v: 0.9, i: 0.7, t: 0.6, o: 0.4, n: 0.3, s: 0.2, @: 2.3}	{x: 97.1, g: 1.2, t: 0.6, @: 1.1}
y	{y: 99.5, @: 0.5}	{y: 98.0, v: 1.1, @: 0.9}	{y: 87.9, j: 3.4, v: 3.1, i: 0.4, r: 0.3, s: 0.2, @: 4.7}	{y: 96.9, v: 1.3, j: 0.3, f: 0.2, @: 1.3}
z	{z: 99.2, s: 0.5, @: 0.3}	{z: 86.0, s: 2.5, x: 1.6, r: 1.2, i: 1.1, a: 0.9, g: 0.3, t: 0.3, v: 0.3, c: 0.2, b: 0.2, e: 0.2, k: 0.2, l: 0.2, o: 0.2, n: 0.2, u: 0.2, @: 4.2}	{z: 68.7, r: 6.2, s: 1.9, b: 1.6, n: 1.6, m: 1.5, y: 1.5, i: 0.8, u: 0.7, l: 0.5, @: 15.0}	{z: 98.1, @: 1.9}

Table 3: Percentages of non-standard mapping 1:*n* (one GT character is substituted by *n* OCR characters). Only values higher than 0.01% are shown. For each GT character, percentages shown for each dataset are parts of corresponding percents of @ in Table 2.

GT Char	Monograph	Periodical	Overproof NLA	Overproof NC
a			{ii: 0.05, in: 0.03, -i: 0.02, .i: 0.02}	{ii: 0.21, it: 0.05, in: 0.05, .i: 0.05, iu: 0.03}
b		{li: 0.19, ti: 0.02, th: 0.02, l.: 0.02}		{’h: 0.11, ili: 0.04}
c		{See: 0.03, foe: 0.02}	{t-: 0.05, e-: 0.04, le: 0.03, i’: 0.02, .e: 0.02}	{Hle: 0.07, ’C: 0.02, iriw: 0.02}
d			{il: 0.15, tl: 0.05, cl: 0.03, ri: 0.03, t4: 0.02}	{il: 0.15, cl: 0.07, rt: 0.06, tl: 0.05, nl: 0.04}
e			{io: 0.04, lc: 0.02, ic: 0.02}	{io: 0.14, iu: 0.03, no: 0.02, oo: 0.02, n;: 0.02}
f			{’l: 0.03, l’: 0.02}	{l’: 0.1, l": 0.05, he: 0.02}
g			{iR: 0.09, a-: 0.08, tr: 0.08, fr: 0.07, er: 0.06}	{i": 0.33, e:: 0.21, uu: 0.14, (;: 0.14, ..:.-: 0.13}
h	{li: 0.07}	{li: 0.78, ii: 0.23, il: 0.07, ri: 0.05, ir: 0.04}	{li: 0.3, il: 0.06, ll: 0.05, ji: 0.02, i(.li: 0.02}	{li: 0.21, di: 0.04, Ii: 0.04, ’li: 0.04, ti: 0.03}
i			{vl: 0.03, ll: 0.02}	{ll: 0.04, l’: 0.02, ’.: 0.02}
j		{.t: 0.08, i.: 0.08}		
k		{lc: 0.06, fc: 0.03}	{lr: 0.12, l;: 0.12, lt: 0.08, fc: 0.06, ’,: 0.04}	
l			{ii: 0.02, uit: 0.02, ->: 0.02}	{’.: 0.05}
m	{rn: 0.36, ni: 0.04, in: 0.03}	{in: 0.17, ra: 0.12, rn: 0.09, ni: 0.08, tn: 0.06}	{in: 0.37, rn: 0.29, ni: 0.13, ra: 0.09, tn: 0.08}	{in: 0.65, ni: 0.48, ro: 0.16, rn: 0.15, tn: 0.11}
n		{r.: 0.07, ri: 0.03, ii: 0.03}	{ii: 0.11, ti: 0.03}	{ii: 0.12, ti: 0.11, ri: 0.08, t.: 0.06, iti: 0.03}
o				{in: 0.03, .i: 0.02, i.: 0.02}
p		{ji: 0.03}	{ii: 0.05, iv: 0.03, .i: 0.02}	{fi: 0.1, iiiti: 0.07, ii: 0.03}
q	{cp: 0.03}	{tj: 0.1, .l: 0.05, ri: 0.05, -’t: 0.05}	{.v: 0.03}	
r			{ii: 0.02, i-: 0.02, li: 0.02, i’: 0.02}	{ii: 0.04, t’: 0.02}
s			{la: 0.03, t,: 0.02, iB: 0.02}	{.-: 0.04, c-: 0.04, nl’: 0.04, i’: 0.04, .": 0.03}
t				{ln: 0.03, Uo: 0.02}
u		{ti: 0.04, ii: 0.02, tt: 0.02, it: 0.02}	{ii: 0.19, ti: 0.08, li: 0.04, tii: 0.03, i.: 0.02}	{ti: 0.11, ii: 0.1, tl: 0.06, ri: 0.05, i’: 0.04}
v			{Ham: 0.09, %’: 0.05, s’: 0.04, «.: 0.02}	{*: 0.24}
w		{vv: 0.03, vr: 0.02, sr: 0.02}	{vv: 0.44, tv: 0.15, ir: 0.07, *v: 0.05, v»: 0.05}	{st: 0.11, fiH: 0.07}
x	{’∼: 0.02}	{ts: 0.03}	{.i: 0.39}	
y			{nj: 0.07, i,: 0.05, ij: 0.05,)*: 0.05, ’j: 0.04}	{tv: 0.04, iiv: 0.04, ino: 0.04, IV: 0.04}
z		{sa: 0.16, .i: 0.16, r.: 0.16, id: 0.16, ti: 0.16}		

Table 4: Percentages of non-standard mapping *n*:1 (*n* GT characters are substituted by one OCR character). Only OCR characters are results of *n* GT characters mis-recognition are listed, and only values higher than 0.01% are shown. Even though this table shows 1:n mapping, the presentation is in a reverse way (1:n) in order to save space.

OCR Char	Monograph	Periodical	Overproof NLA	Overproof NC
a	{whe: 0.03, we: 0.02, The: 0.02}	{ste: 0.07, ur: 0.05, pe: 0.05, our: 0.04, es: 0.04, co: 0.02, nc: 0.02, us: 0.02, ec: 0.02, et: 0.02, ly: 0.02}	{s.: 1.69, s,: 0.36, ce: 0.09, ut: 0.07, en: 0.03, nd: 0.03, er: 0.03}	{si: 0.55, he: 0.07}
b	{li: 0.03}	{li: 0.08, ch: 0.02, el: 0.02, le: 0.02, th: 0.02}	{hi: 0.07, li: 0.06, is: 0.05}	{si: 0.21}
c	{pe: 0.05}	{el: 0.04, ea: 0.03, pe: 0.02, es: 0.02, rs: 0.02}	{e,: 0.47, le: 0.13, ee: 0.12, ne: 0.12, er: 0.03, ng: 0.02}	{ess: 0.36, es: 0.25, ee: 0.18, se: 0.1, le: 0.02}
d	{il: 2.62, ll: 0.45, ill: 0.02}	{il: 0.1, el: 0.06, ll: 0.04, ct: 0.03, rt: 0.02, al: 0.02, ti: 0.02}	{il: 0.08, ol: 0.03}	{si: 0.24, on: 0.08}
e	{ho: 0.04, oun: 0.02}	{io: 0.02}	{s.: 0.12, ic: 0.07, ol: 0.03}	{can: 1.31, ic: 0.57, ac: 0.43, his: 0.27, ct: 0.02}
f		{wa: 0.02}	{ta: 0.13}	
h	{li: 0.28, la: 0.02}	{li: 0.08, la: 0.06, is: 0.04, le: 0.04, si: 0.02}	{ly: 0.38, li: 0.26, s.: 0.04}	{ld: 0.12}
i	{wa: 0.02}	{ac: 0.03, ll: 0.03, ea: 0.03, ec: 0.02, ra: 0.02, pe: 0.02, ho: 0.02}	{r.: 3.46, s.: 0.39, nce: 0.38, al: 0.05, as: 0.05, st: 0.04, ha: 0.04, er: 0.04, nd: 0.03, at: 0.02}	{ta: 0.26, on: 0.05}
j		{ie: 0.03, ee: 0.02, la: 0.02}	{or: 0.05}	
k		{le: 0.04, ic: 0.03, io: 0.03, is: 0.02}	{ly: 0.22}	
l		{ir: 0.03, ie: 0.03, is: 0.02}	{ni: 0.26, ri: 0.19, si: 0.18, di: 0.14, r.: 0.07, st: 0.02}	{ir: 1.02, ai: 0.6, ot: 0.21, in: 0.12, re: 0.06}
m	{in: 0.07, ste: 0.03, ra: 0.02}	{us: 0.15, ns: 0.11, un: 0.1, in: 0.1, res: 0.08, nt: 0.08, ur: 0.05, ss: 0.05, ver: 0.04, ee: 0.04, ra: 0.04, ne: 0.03, rs: 0.02, an: 0.02, ar: 0.02, re: 0.02, si: 0.02, io: 0.02, co: 0.02}	{n,: 0.43, ur: 0.3, ni: 0.29, in: 0.29, ia: 0.25, ns: 0.16, ai: 0.15, ree: 0.12, as: 0.07, rs: 0.05, or: 0.04, ou: 0.03, an: 0.03, on: 0.02, ra: 0.02, he: 0.02}	{ld: 0.55, ns: 0.42, ll: 0.21, nt: 0.11, es: 0.09, ee: 0.08, on: 0.05}
n	{ri: 0.22, ll: 0.02, ra: 0.02}	{ri: 0.24, rs: 0.14, us: 0.05, wh: 0.05, rt: 0.04, ll: 0.04, il: 0.04, Th: 0.03, ro: 0.03, ss: 0.02, ut: 0.02, re: 0.02, as: 0.02, at: 0.02, li: 0.02, ic: 0.02, is: 0.02}	{ri: 1.61, ry: 0.54, ia: 0.54, am: 0.23, ma: 0.13, ra: 0.13, s.: 0.12, s,: 0.12, st: 0.12, ll: 0.11, ti: 0.1, ay: 0.08, ar: 0.05, at: 0.05, er: 0.04, il: 0.03}	{rs: 0.99, ss: 0.47, om: 0.41, as: 0.28, ar: 0.05, es: 0.03}
o	{el: 0.02}	{el: 0.04, ay: 0.03, ee: 0.02, si: 0.02, se: 0.02}	{e,: 0.75, ic: 0.35, ie: 0.27, nc: 0.23, ne: 0.13, me: 0.12, es: 0.09, ive: 0.07, he: 0.07}	{ee: 0.32, se: 0.3, ll: 0.15, es: 0.14, en: 0.08, re: 0.03}
p			{ve: 0.12, s,: 0.12, ing: 0.1, on: 0.02}	
q			{s.: 0.27}	{ar: 0.1}
r		{ot: 0.02, ve: 0.02, la: 0.02}	{ac: 0.23, ss: 0.12, ee: 0.09, ce: 0.03, he: 0.03}	{me: 0.21, en: 0.18}
s		{ear: 0.06, ta: 0.02}	{e,: 0.14, ng: 0.07, he: 0.03}	{tor: 1.99, an: 0.08}
t		{be: 0.04, il: 0.04, ce: 0.03, ge: 0.03, ie: 0.03, nc: 0.02, si: 0.02, li: 0.02, ra: 0.02}	{ine: 0.6, e,: 0.29, one: 0.22, s.: 0.16, le: 0.13, er: 0.04, nd: 0.03}	
u	{oo: 0.02, we: 0.02, ir: 0.02, il: 0.02}	{ss: 0.12, as: 0.1, ta: 0.08, nde: 0.06, ie: 0.06, il: 0.04, ns: 0.04, is: 0.03, tr: 0.03, ne: 0.03, ri: 0.03, ai: 0.03, rt: 0.02, ec: 0.02, li: 0.02, ee: 0.02, io: 0.02, si: 0.02}	{is: 0.37, ri: 0.28, so: 0.27, ia: 0.25, ll: 0.25, rs: 0.19, as: 0.19, hi: 0.16, ti: 0.13, il: 0.12, ha: 0.11, le: 0.1, in: 0.07, ra: 0.06, st: 0.06, li: 0.06, ee: 0.05, ai: 0.04, re: 0.03, it: 0.02, he: 0.02}	{ns: 0.7, na: 0.65, fo: 0.24, st: 0.2, an: 0.15, ea: 0.14, te: 0.06, is: 0.06}
v		{ai: 0.03}	{ry: 0.04}	
w	{hav: 0.03}	{ss: 0.19, ec: 0.05, se: 0.05, ar: 0.04, tr: 0.03, ea: 0.03, co: 0.02, ve: 0.02, un: 0.02, ur: 0.02, ee: 0.02, si: 0.02, fo: 0.02, ta: 0.02}	{si: 0.11, or: 0.02}	{ear: 1.08, se: 0.29}

of *single-error* typos in misspelled words (74.5% on average) [13]. In terms of *multi-error* tokens, most of them are of edit distance 2 (on average 22.57%). These statistics reveal that OCR post-processing approaches can mainly concentrate on edit distances 1 and 2 (with total 81.49% on average) at beginning steps. Relying on these statistics, the edit distance threshold can be set at 2 for removing many irrelevant candidates.

4.1.5 String similarity based on Longest Common Sequence (LCS). LCS is another way to measure the similarity between two strings. Islam *et al.* [9] proposed two variations of LCS, including Normalized Longest Common Subsequence (NLCS) and Normalized Maximal Consecutive Longest Common Subsequence (NMCLCS). NLCS considers lengths of two related strings, as follows:

$$NLCS(w_c, w_e) = \frac{len(LCS(w_c, w_e))^2}{len(w_c) * len(w_e)} \qquad (1)$$

There are three variations of MCLCS (Maximal Consecutive Longest Common Subsequence) with some additional conditions. $MCLCS_1$ and $MCLCS_n$ use MCLCSs beginning at the first, and at the n-th character, respectively; $MCLCS_z$ only considers MCLCSs ending at the last character.

$$NMCLCS_i(w_c, w_e) = \frac{len(MCLCS_i(w_c, w_e))^2}{len(w_c) * len(w_e)} \qquad (2)$$

where $MCLCS_i$ can be $MCLCS_1$, $MCLCS_n$ or $MCLCS_z$.
The similarity of the two strings S is calculated as below:

$$S(w_c, w_e) = \alpha * NLCS(w_c, w_e) + \sum_{i \in \{1, n, z\}} \alpha_i * NMCLCS_i(w_c, w_e) \qquad (3)$$

where α, α_i are weights of NLCS and $NMCLCS_i$.
We reuse the same weights suggested by Islam *et al.* [9] in our statistics. Fig. 3 shows rates of errors on the four datasets with different threshold values of similarity S. Our observation reveals that about 83.5% of all errors have the similarity S equal or greater than 0.125. Similar to edit distance, the threshold of LCS similarity can be used in removing many incorrect candidates for each error.

4.2 Length effects

As to length effects, we examine not only word lengths but also OCR token lengths. Furthermore, we suggest a novel classification by grouping errors according to word lengths and edit distances.

4.2.1 Word length. In terms of word length, Kukich [13] found that more than 63% of the spelling errors are *short-word* errors.

Percentages of correct/incorrect word recognition according to word lengths on our datasets are shown in Fig. 4. According to our statistics, about 42.1% of OCR errors are *short-word* errors, which

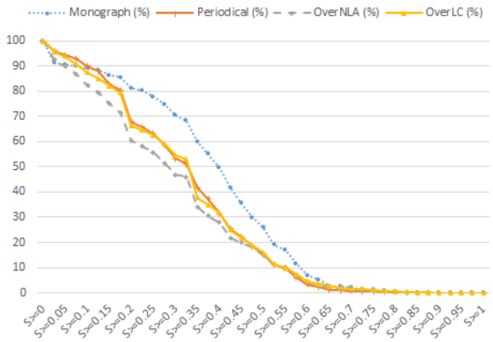

Figure 3: Error rates based on the LCS similarity S

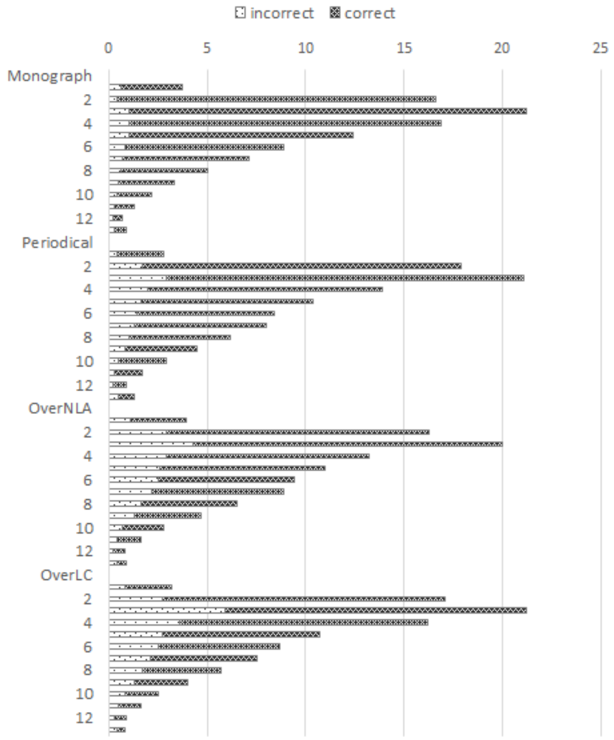

Figure 4: Rates of correct and incorrect word recognition based on word lengths

is a lower value than that of misspellings with 63% on average. In addition, from the highest percentage at length 3, the percentage of incorrect word recognition decreases gradually according to the increase of GT token length. Furthermore, around 85.27% of all OCR errors occur in words of lengths from 2 to 9.

4.2.2 OCR token length. In practice, post-OCR approaches have to deal with OCR tokens instead of GT words, and lengths of OCR tokens can differ from those of GT words, therefore we consider lengths of OCR tokens. For example, in OCR tokens *'scho ol'* and their GT word *'school'*, two incorrect OCR tokens are *'scho'* of length 4, and *'ol'* of length 2; these OCR tokens come from GT word of length 6.

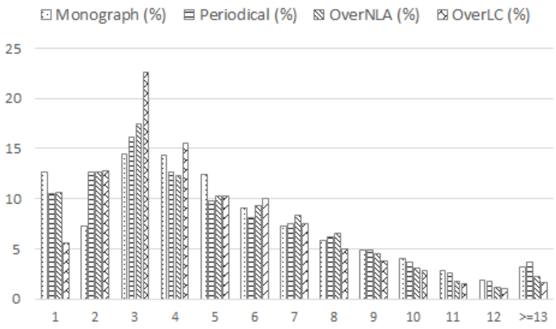

Figure 5: Error rates based on OCR token lengths

Similar to word length, the analysis of incorrect OCR token lengths (see Fig. 5) suggests that incorrect OCR tokens of length 3 are the most common one. In addition, about 80.55% of all invalid OCR tokens are of lengths between 2 and 9.

4.2.3 Two-dimensional classification based on word lengths and edit distances. There are some arguments that it is more difficult to deal with *short-word* errors than with errors appearing in longer-length words. Because *short-word* errors are more likely to yield another lexicon entry when applying character edit operations [14].

However, we believe that the problem does not only result from length but also from edit distance between an error and its GT word. For example, there are two errors (e.g. *'ict'*, *'lct'*) and their GT word (e.g. *'let'*). The first error *'ict'* requires 2 edit operations to be transformed into its GT word, which is more challenging than the second error *'lct'* needing only 1 modification to be converted to its GT word. To give a clear view of such problem, we suggest a novel classification by grouping errors according to word lengths and edit distances. With run-on errors (e.g. *'blue sky'* vs. *'blucsky'*), we assume the sum of lengths of words related to the errors as their word length.

The two-dimensional classification of four datasets is shown in Fig. 6. Based on this classification, some post-processing approaches can decide edit distance thresholds for each word length. As mentioned in Sec 4.1.4, around 81.49% of errors have edit distance of 1, 2. In other words, maximum number of possible errors that post-processing approaches can correct is about 81.49% if edit distance threshold is set as 2 for all word lengths.

In our opinion, by adjusting edit distance threshold according to word length, post-OCR techniques can deal with higher rate of errors. Based on our observations, we suggest to set edit distance thresholds 2, 3, 4 for word lengths less than 4, 10, 13, respectively. On average, those settings increase the rate of errors that post-OCR techniques can process from 81.49% to 89.15%.

4.3 Erroneous character positions

The survey on misspellings [13] has shown that there are a few errors at the 1st character. However, there is no research related to erroneous character positions in OCR text. Hence, we examine OCR errors at different character positions, including the first/last/middle

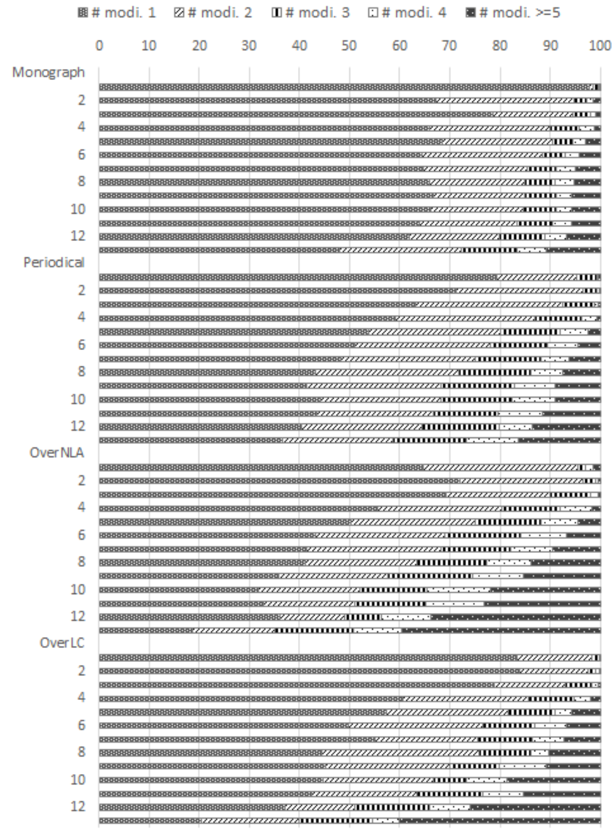

Figure 6: Error rates based on word lengths and edit distances

position (denoted as *first, last, nth,* respectively), and their possible combinations (denoted as *first+last, first+nth, last+nth, first+last+nth* respectively). In case of *run-on* errors, because this error type incorrectly removes white space at the end of the first word, we decide that this error type always has one *last-position* error.

Details of erroneous character positions of our datasets are shown in Fig. 7. While 12.46% of OCR errors are *first-position* errors, spelling errors have slightly smaller percent of such errors with average 11% of all errors.

It is noticeable that on average 27.37% of all errors are *last-position* errors, which are even comparable with that of *middle-position* errors (28.69%). Moreover, our observations on four datasets indicate that erroneous characters rarely appear at the first/last position in the same error. In fact, statistics show that less than 10% of errors belong to (*first + last*) or (*first + last + nth*) combinations. Therefore, OCR post-processing can firstly focus on single positions or some combinations (*first+nth, last+nth*).

4.4 *Real-word* vs. *non-word* errors

In the next analysis we study the rate of *real-word* and *non-word* errors in OCR text. *Real-word* errors are valid in dictionary but incorrect in context (e.g. *'hear'* vs. *'bear'*). The amount of *real-word* errors varies naturally with the size of the lexicon [21]. Too small lexicon can ignore valid tokens and increase the number of false negatives. In contrast, a too large dictionary can match invalid tokens to low-frequent lexical entries or special domain terms, potentially raising the number of false positives. In other words, the larger the lexicon is, the more *real-word* errors can occur.

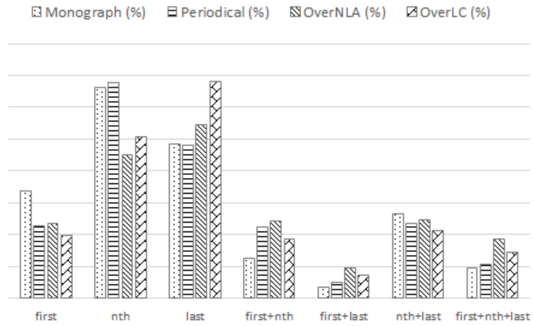

Figure 7: Error rates of erroneous character positions

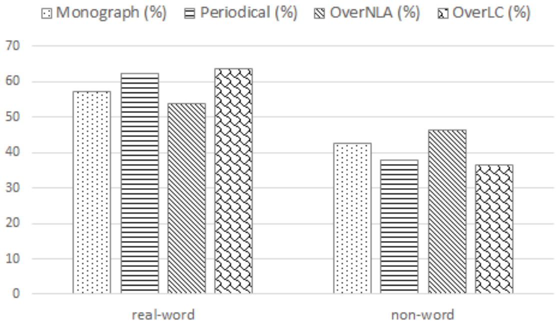

Figure 8: Rates of real-word vs. non-word errors

On the other hand, *non-word* errors are invalid in dictionary (e.g. *'hear'* vs. *'hcar'*). It is obvious that *non-word* errors are easier to be detected and corrected than *real-word* errors. In addition, there are words which appear in GT but are not lexicon entries, known as out-of-vocabulary (OOV) words. Using the word frequency of COHA corpus, the rate of OOV words in our datasets is found to be about 1%.

The statistics of *real-word* errors and *non-word* errors in Fig. 8 show that approximately 59.21% of OCR errors are *real-word* errors. The proportion of *real-word* errors in our four datasets is about 1.47 times higher than that of *non-word* ones. On the contrary, misspellings have opposite trend with 67.5% *non-word* errors.

Our observations on the four datasets also indicate that approximately 13.77% of non-word errors involve digits, and 25.08% of real-word errors relate to punctuations. High percentage of punctuation errors is one notable feature of OCR text. In fact, the low physical quality of old documents causes misrecognition of punctuation. Therefore, OCR texts tend to contain more incorrect/redundant commas and dots than human-generated texts.

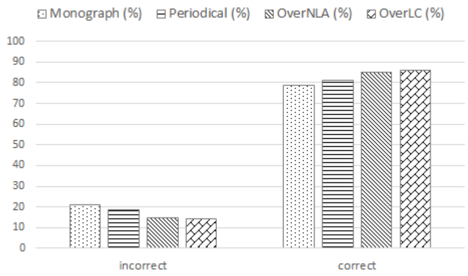

Figure 9: Rates of correct vs. incorrect word boundary errors

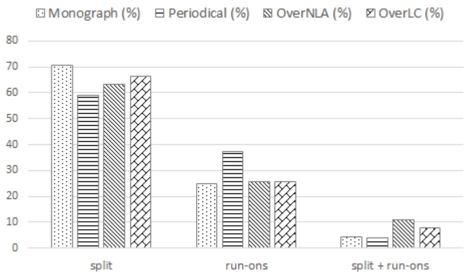

Figure 10: Error rates of incorrect word boundary subtypes

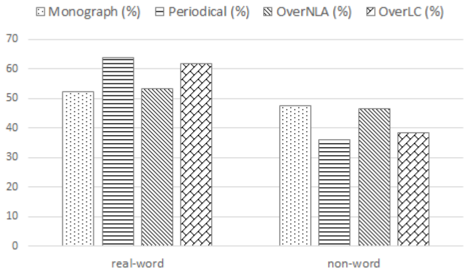

Figure 11: Error rates of correct word boundary subtypes

4.5 Word boundary

This subsection observes word boundary aspect in detail. Let us call errors related to wrongly identified word boundaries as incorrect word boundary errors, and ones unrelated to word boundary problems as correct word boundary errors.

To give clearer views of OCR errors, we suggest to make a hierarchical classification based on incorrect/correct word boundary error types, and *real-word*/*non-word* error types. We firstly separate OCR errors into incorrect/correct word boundary error types. Secondly, in terms of incorrect word boundary error type, depending on inserting/deleting white spaces we classify into two main sub-types, including *incorrect split*/*run-on* error types. In terms of correct word boundary error type, we divide into *real-word*/*non-word* error types. Finally, for *incorrect split*/*run-on* error types we continue grouping into *real-word*/*non-word* error types.

Percentages of incorrect/correct word boundary types of our four datasets are shown in Fig. 9. It is clear that all of the four datasets give a similar trend. Around 82.85% of errors are correct word boundary errors, which is much higher than that of incorrect word boundary ones.

4.5.1 *Incorrect word boundary errors.*

In terms of incorrect word boundary errors, we study two popular sub-types: *incorrect split*/*run-on* error types.

Incorrectly putting two or more words together creates a *run-on* error which is often not in the lexicon. In other words, most of *run-on* errors are *non-word* errors, and they are easy to be detected. Correcting such errors is more complicated because it easily leads to a combinatorial explosion of the number of possible word combinations.

Wrongly splitting one word into some strings results in *incorrect split* errors. Both detecting and correcting such errors are challenging because some of split strings are not in the lexicon (*non-word* errors) and others are lexicon entries (*real-word* errors).

Percentages of incorrect word boundary sub-types of four datasets are shown in Fig. 10 with *incorrect split* errors denoted as *split*, *run-on* errors denoted as *run-on* and their combination (*split + run-on*). It is notable that the percent of *incorrect split* errors is on average 2.36 times higher than that of *run-on* errors. In contrast, most of incorrect word boundary errors in misspellings are *run-on* errors with 6.5 times higher occurrence than *incorrect split* ones. In addition, *incorrect split* and *run-on* errors rarely appear together in errors. The percentage of their combination (*split + run-on*) is only 6.8% on average, therefore, post-processing approaches can ignore it at first steps.

4.5.2 *Correct word boundary.*

In terms of correct word boundary, we directly classify errors into *real-word*/*non-word* error types. Percentages of *real-word* and *non-word* errors in correct word boundary type are shown in Fig. 11. Real-word/non-word errors mentioned in this section are sub-sets of the *real-word* and *non-word* errors pointed out in Fig. 8, which reveals the similar trend with their super-sets. Other *non-word* and *real-word* errors are of the incorrect word boundary type, with 28.72% *real-word* errors and 24.82% *non-word* ones, on average.

5 SUMMARY OF MAIN FINDINGS

We summarize in this section the key observations from our study. Firstly, we examined OCR errors and compared them with spelling errors in several aspects. Misspellings and OCR errors have similar trends in two cases. In particular, most of them are *single-error* errors (74.5% misspellings, 58.92% OCR errors), and few of them are *first-position* errors (11% misspellings, 12.46% OCR errors).

However, misspellings and OCR errors differ in three other aspects, including *real-word* vs. *non-word* errors, *incorrect split* vs. *run-on* errors, *short-word* errors. We found that most of misspellings (67.5%) are *non-word* errors while most of OCR errors (59.21%) are *real-word* ones. Regarding the incorrect word boundary error type, the percentage of *run-on* errors is 6.5 times higher than that of *incorrect split* ones in case of spelling errors. In contrast, the proportion of *incorrect split* errors is on average 2.36 times greater than that of *run-on* errors in case of OCR errors. Moreover, while 63% of misspellings appear in short words, only 42.1% of OCR errors are *short-word* errors.

Secondly, besides similar aspects as in Kukich's survey, we present novel statistics (non-standard mappings, string similarities based on LCS, OCR token lengths, and erroneous character positions). For non-standard mappings, our analysis reveals that some characters *‘b’, ‘d’, ‘h’, ‘m’, ‘n’* are easily recognized as *‘li’*, {*‘il’, ‘cl’*}, *‘li’*, {*‘rn’, ‘in’*}, {*‘ri’, ‘ii’*}, respectively. In opposite way, some strings *‘li’*, {*‘il’, ‘cl’*}, *‘li’*, {*‘rn’, ‘in’*}, {*‘ri’, ‘ii’*} can be recognized as *‘b’, ‘d’, ‘h’, ‘m’, ‘n’*, respectively. In case of string similarities based on LCS, around 83.5% of OCR errors achieve no less than 0.125 similarity S with their GT words. As to OCR token lengths, they show similar trend with word lengths. Particularly, incorrect OCR token of length 3 is the most common, and most of erroneous OCR tokens are of lengths from 2 to 9.

For erroneous character positions, around 27.37% errors are *last-position* errors, and they thus are comparable to *middle-position* errors (28.69%). In addition, we observe that errors rarely have errorenous characters at both the first and last position (in total 9.75% of *first+last* and *first+last+nth*).

Finally, based on the analysis on four datasets, we make some suggestions for designing post-processing approaches. Because *last-position* errors rarely appear together with *first-position* errors, post-OCR techniques can ignore their combinations (*first+last*, *first+last+nth*).

Our observations show that deletion, insertion and substitution occasionally appear together in the same word (around 22.98%); algorithms of candidate generation can then pay more attention on single modification types instead of their combinations. Moreover, the rate of the number of substitution/deletion/insertion character candidates for each character position of OCR token can be set as 5:1:1 in generating candidates.

Edit distance is considered as an important criteria in selecting relevant candidates. Interestingly, 81.49% of OCR errors are of edit distance 1 or 2, so with edit distance threshold 2, post-processing approaches could easily remove many irrelevant candidates. Moreover, edit distance thresholds can be adjusted according to word lengths. With flexible settings of edit distance threshold, post-processing techniques would be able to handle about 89.15% of errors.

6 CONCLUSION

In this paper, we examine different aspects of OCR errors towards a better understanding of OCR errors and related challenges. Based on our observations on four datasets we also suggest guidelines for designing post-processing approaches. In addition, we propose a novel two-dimensional classifications, including grouping errors according to word lengths and edit distances, as well as grouping of *real-word*/*non-word* errors following word boundary types. Our work can be viewed as an important, initial step to further analyses or towards more efficient and robust post-OCR techniques.

ACKNOWLEDGMENTS

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant 770299 (NewsEye) and in part by MEXT Grant-in-Aid (#18H03243).

REFERENCES

[1] Mark A. Jones, Guy A. Story, and Bruce W. Ballard. 1991. Interating multiple knowledge sources in a bayesian ocr post-processor. *International Journal on Document Analysis and Recognition* (1991), 925–933.

[2] Haithem Afli, Loïc Barrault, and Holger Schwenk. 2016. OCR Error Correction Using Statistical Machine Translation. *Int. J. Comput. Linguistics Appl.* 7, 1 (2016), 175–191.

[3] Youssef Bassil and Mohammad Alwani. 2012. Ocr post-processing error correction algorithm using google online spelling suggestion. *arXiv preprint arXiv:1204.0191* (2012).

[4] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. ICDAR2017 Competition on Post-OCR Text Correction. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, Vol. 1. IEEE, 1423–1428.

[5] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, Muriel Visani, and Jean-Philippe Moreux. 2017. Impact of OCR errors on the use of digital libraries: towards a better access to information. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*. IEEE Press, 249–252.

[6] Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* 7, 3 (1964), 171–176.

[7] John Evershed and Kent Fitch. 2014. Correcting noisy OCR: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*. ACM, 45–51.

[8] Paul Hagon. 2013. Trove crowdsourcing behaviour. *Australian Library & Information Association Information Online 2013 Proceedings* (2013).

[9] Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using Google Web IT 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 1241–1249.

[10] Ido Kissos and Nachum Dershowitz. 2016. Ocr error correction using character correction and feature-based word classification. In *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 198–203.

[11] Okan Kolak, William Byrne, and Philip Resnik. 2003. A generative probabilistic OCR model for NLP applications. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. 55–62.

[12] Karen Kukich. 1992. Spelling correction for the telecommunications network for the deaf. *Commun. ACM* 35, 5 (1992), 80–90.

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

[14] Thomas K Landauer and Lynn A Streeter. 1973. Structural differences between common and rare words: Failure of equivalence assumptions for theories of word recognition. *Journal of Memory and Language* 12, 2 (1973), 119.

[15] Rafael Llobet, Jose-Ramon Cerdan-Navarro, Juan-Carlos Perez-Cortes, and Joaquim Arlandis. 2010. OCR post-processing using weighted finite-state transducers. In *2010 International Conference on Pattern Recognition*. IEEE, 2021–2024.

[16] Jie Mei, Aminul Islam, Yajing Wu, Abidalrahman Moh'd, and Evangelos E Milios. 2016. Statistical learning for OCR text correction. *arXiv preprint arXiv:1611.06950* (2016).

[17] Roger Mitton. 1987. Spelling checkers, spelling correctors and the misspellings of poor spellers. *Information processing & management* 23, 5 (1987), 495–505.

[18] George Nagy, Thomas A Nartker, and Stephen V Rice. 1999. Optical character recognition: An illustrated guide to the frontier. In *Document Recognition and Retrieval VII*, Vol. 3967. International Society for Optics and Photonics, 58–70.

[19] Thi-Tuyet-Hai Nguyen, Mickaël Coustaty, Antoine Doucet, Adam Jatowt, and Nhu-Van Nguyen. 2018. Adaptive Edit-Distance and Regression Approach for Post-OCR Text Correction. In *Maturity and Innovation in Digital Libraries - 20th International Conference on Asia-Pacific Digital Libraries*. 278–289.

[20] Hisao Niwa and Kazuhiro Kayashima. [n. d.]. Postprocessing for Character Recognition Using Keyword Information.

[21] James L Peterson. 1986. A note on undetected typing errors. *Commun. ACM* 29, 7 (1986), 633–637.

[22] Michael Piotrowski. 2012. Natural language processing for historical texts. *Synthesis lectures on human language technologies* 5, 2 (2012), 1–157.

[23] Martin Reynaert. 2005. *Text-induced spelling correction.* Ph.D. Dissertation. Tilburg University.

[24] Martin Reynaert. 2008. Non-interactive OCR post-correction for giga-scale digitization projects. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 617–630.

[25] Martin WC Reynaert. 2011. Character confusion versus focus word-based correction of spelling and OCR variants in corpora. *International Journal on Document Analysis and Recognition (IJDAR)* 14, 2 (2011), 173–187.

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

[27] Xiang Tong and David A Evans. 1996. A statistical approach to automatic OCR error correction in context. In *Fourth Workshop on Very Large Corpora*.

[28] Charlene W Young, Caroline M Eastman, and Robert L Oakman. 1991. An analysis of ill-formed input in natural language queries to document retrieval systems. *Information Processing & Management* 27, 6 (1991), 615–622.